EMANUEL LAW FIRM
Sacha V. Emanuel (State Bar No. 218705)
A Professional Corporation
svemanul@live.com
1875 Century Park East, Suite 2150
Los Angeles, California 90067
Telephone: (310) 881-6814
Facsimile: (310) 881-6801

Attorneys for Plaintiff/Counter-Defendant
Neofonie GmbH

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

NEOFONIE GMBH, a German corporation,

Plaintiff,

vs.

ARTISSIMO DESIGNS LLC, a Delaware limited liability company,

Defendant.

AND RELATED COUNTER-CLAIM

Case No.: CV-00772-CJC-JDE

DISCLOSURE OF EXPERT TESTIMONY PURSUANT TO RULE 26(a)(2)(A), F.R.Civ.P.

Complaint Filed:	May 2, 2017
Trial Date:	January 22, 2019

TO DEFENDANT/COUNTER-CLAIMANT AND ITS COUNSEL OF RECORD:

PLEASE TAKE NOTICE THAT plaintiff/counter-defendant, NEOFONIE GMBH, shall present evidence at trial pursuant to Rules 702, 703 or 705, F.R.Ev., of the following persons:

Dr. Ali Khoshgazaran, Ph.D., 205 S Broadway, Suite 300, Los Angeles, CA 90012.

The testimony of Mr. Khoshgazaran shall be in accordance with the attached report prepared concurrently herewith.

DATED: September 18, 2018

EMANUEL LAW FIRM

By: Sacha V. Emanuel
Sacha V. Emanuel
Attorneys for Plaintiff/Counter-Defendant Neofonie GmbH

IN THE UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

NEOFONIE GMBH, a German corporation,

Plaintiff**,**

vs.

ARTISSIMO DESIGNS LLC, a
Delaware limited liability company,

Defendant.

Case No. 8:17-cv-00772 CJC (JDEx)

EXPERT REPORT OF ALI KHOSHGOZARAN, PHD

TABLE OF CONTENTS

1 INTRODUCTION AND SCOPE OF ENGAGEMENT

1. I have been retained as an expert in computer science in connection with the matter of *Neofonie GMBH v. Artissimo Designs LLC* to provide analysis and opinions related to the design and development of artdesigns.com (the Website) by Neofonie for Artissimo. Specifically, I was asked to provide a quantitative and qualitative analysis of Neofonie's work and the process it used to develop the Website as well as identify the root cause of delays in launching the Website.

I submit this expert report as a statement of the opinions that I have formed and the factual basis for those opinions. If necessary, I am prepared to testify at trial as to the matters discussed in this report. Between now and such time that I may be asked to testify, I expect to continue my review, evaluation, and analysis of information generated during discovery. I also expect to review relevant evidence presented before and/or after trial. I also understand that additional material may be made available that is relevant to my analysis. I also expect to review the reports submitted by Defendant's experts. I reserve the right to amend or supplement this report, as necessary and as acceptable to the court. I also reserve the right to develop materials and exhibits as appropriate for use in helping to demonstrate and explain my opinions in the event that I am asked to testify at trial.

1.1 Qualifications

2. This section summarizes my education, professional achievements, and peer-reviewed scientific publications in the field of computer science. A more detailed list of my qualifications is set forth in my curriculum vitae, a copy of which is attached hereto as Exhibit A.

1.1.1 Education

3. I graduated with a Bachelor of Science degree in computer engineering from Sharif University of Technology in 2003. My undergraduate degree in software engineering, focused on

principles and advances in teaching software engineering to computer science undergraduate students in premier academic institutions. As an undergraduate, I was also a member of a software engineering lab conducting research in software engineering's best practices. In 2005, I received my Master of Science degree in computer science from The George Washington University where I spent two years conducting research on establishing a framework for statistical cryptanalysis.

4. I also attended graduate school at the University of Southern California, where I was a USC Viterbi School of Engineering graduate research and teaching assistant. I received my PhD in computer science in 2010 from USC. My PhD research focused on big data and privacy aspects of spatio-temporal information management. My research was funded by several government agencies and technology companies such as The National Science Foundation (NSF), Microsoft and Google. I have published my findings in numerous peer-reviewed conference papers, magazines, book chapters, journal articles and other publications. At USC I also led a team of software engineers to develop GeoDec (Geo-spatial Decision-Making), a platform to rapidly and accurately build an information-rich and realistic geospatial spaces (e.g., a city) with a temporal dimension while supporting visualization, complex querying and data analysis capabilities.

1.1.2 Employment

5. I am currently employed as a computer scientist at Quandary Peak Research, Inc. in Los Angeles, CA. In this role, I analyze the development, structure, behavior, and quality of software systems. I have analyzed a broad variety of complex, real-world software systems. I have reverse-engineered the software designs of several of software systems. I have conducted many investigations of the design and implementation of these systems with respect to specific patent claims and other intellectual property considerations.

6. I formerly worked at Yahoo, Microsoft and Samsung. During my time with Yahoo, I developed tools for Yahoo's behavioral targeting team to query, analyze and visualize massive amounts of user behavior data while interacting with 300+ Yahoo properties. A patent was filed with Yahoo on technologies developed. As a product manager at Microsoft, I was responsible for the software specifications of a key application for Microsoft's Bing Maps platform to ingest, analyze and visualize third-party event data on maps. At Samsung, I led a team of software developers to build and release TV applications using SyncPlus, Samsung's Automatic Content Recognition platform. I worked closely with Samsung key partners (Verizon, Turner Broadcasting and Time Warner Cable) to launch their connected TV applications in the US market.

7. After leaving Samsung, I co-founded Tilofy, a Software as a Service platform for trend forecasting. Tilofy's proprietary trend forecasting technology employs advances in information management and artificial intelligence to predict trends in their infancy, long before they reach the mainstream. While leading Tilofy, the company raised $2M in venture financing and signed up some of the world's prominent brands (Red Bull, Samsung, RJ Reynolds, Hershey, wipro digital, Unilever and many more) as paid customers of Tilofy's trend forecasting platform. In 2014, I was awarded the prestigious SoCalTech50: Southern California's Top 50 People to Watch in Tech Industry along with CEOs of Tinder, Whisper and Snap.[1]

8. In my career I have built numerous award-winning and patented consumer-facing and enterprise products for mobile, television and desktop platforms that have been featured in various news outlets such as Forbes, Huffington Post, United Nations System Staff College, ABC7, and NBC.

[1] Announcing the 2014 socalTECH 50: Southern California's Ones To Watch
https://www.socaltech.com/announcing_the___4_socaltech___southern_california_s_ones_to_watch/s-0057749.html

1.1.3 Publications

9. I have authored over fifteen scholarly journal articles, magazine articles, conference papers, and book chapters on varied topics related to computer science, software engineering and spatio-temporal information systems. I have delivered numerous presentations, seminars, and technology demonstrations related to software design and analysis for research symposia, conferences, workshops, and industry events in the field of computer science. I have also served as a reviewer or panelist for over a dozen computer science journals, magazines, and conferences.

10. I have also frequently served as a speaker, judge or panelist at various industry events and have given numerous guest lectures in a wide range of computer science topics at The George Washington University, the University of Southern California and California State Polytechnic University of Pomona.

11. I have never been deposed or testified in trial before.

1.2 Compensation

12. Quandary Peak Research is being compensated for my work on this case at the rate of $275 per hour plus reimbursement of direct expenses. I have no personal interest in this litigation, and my compensation does not depend in any way on the opinions I express or outcome of this case.

2 BASIS FOR OPINIONS AND MATERIALS CONSIDERED

13. My opinions in this report are based upon at least the following: (1) my years of experience as a technology executive and as a computer scientist; (2) relevant documents produced in this case; (3) scientific literature, authoritative texts, and other references in the field of the claimed technologies, where cited in this report.[2] Attached to this report as Exhibit B, I have listed the materials I reviewed and analyzed in the course of forming my opinions.

[2] I used Google Translate plugin for Chrome in order to translate German content present in web pages such as contents of JIRA tickets and other relevant case documents that contained German text.

3 OVERVIEW OF WORK PERFORMED BY NEOFONIE

14. I reviewed several documents to understand the nature of work Defendant and Plaintiff agreed to be performed by Plaintiff. I understand that Plaintiff, an e-commerce solution provider based in Germany, was hired by Defendant to design, develop and launch an e-commerce website to enable Defendant to capture market opportunities in Wall Décor and further expand its product offerings into the premium category. In particular, I studied the document titled "Offer Development, Test and Launch of a MVP for artdesigns.com" which contains the agreement between Defendant and Plaintiff to develop and launch a minimum viable product (MVP) for the Website. In the rest of my report, I refer to this document as Phase 2 Offer.

3.1 Key Timeline of Events

15. Upon reviewing case documents, I identified several key dates that bear significance to this case. The dates, along with the documents I used to identify each date, are listed below.

- Initial presentation to Artissimo: Feb 22, 2016 (Neofonie presentation).
- Completion of phase 1: May 6, 2016 (Phase 2 Offer).
- Initial agreement for MVP: May 23, 2016 (Phase 2 Offer).
- Initial anticipated launch date: Sep 19, 2016. (Defendant's Answer and Counterclaim).
- First extension: Oct 10, 2016. (Defendant's Answer and Counterclaim).
- Second extension: Oct 30, 2016. (Defendant's Answer and Counterclaim).
- Completion of development: Nov 7, 2016 (Plaintiff's email to Defendant).
- Decision to cancel the project: Nov 23, 2016 (Defendant's Answer and Counterclaim).

3.2 Phase 1. Identification of MVP and Creation of a Backlog

16. I reviewed the presentation titled "artdesigs.com eCommerce Solution" dated Feb 22, 2016 that provides a background on Plaintiff's relevant prior work as well as a proposal for phase 1 of the project. According to this document, phase 1, which is named "Specification", was planned to include capturing requirements, creating the high-level information architecture, and developing

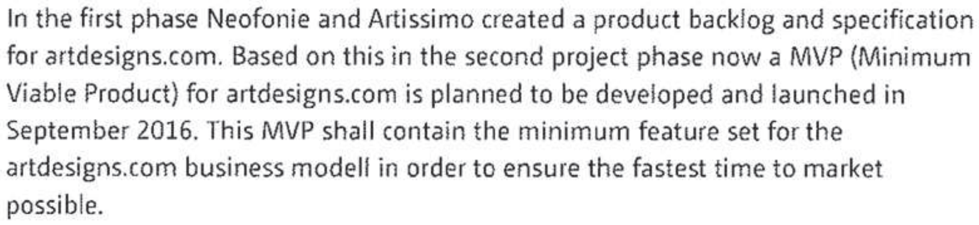
the concept for the Website as well as the user experience (UX) and user interface (UI) design work. Plaintiff's assessment and proposal for phase 1 indicated 62 days for completion of this phase.

17. As part of phase 1 of the project, Plaintiff agreed to create a product backlog that reflects the specification and concept for the Website. In addition, in order to capture what is needed to build the website, identification of stakeholders and high-level design of key screens were also included as part of phase 1 of the project.

3.3 Phase 2. Implementation of MVP and Go Live

18. The Phase 2 Offer reflects the mutual agreement between Plaintiff and Defendant to develop, test and launch a Minimum Viable Product (MVP) to provide an effective way for Defendant to test various hypotheses of customer needs and user behavior to be used for further development.

Figure 1. Scope of MVP as Stated in Phase 2 Offer

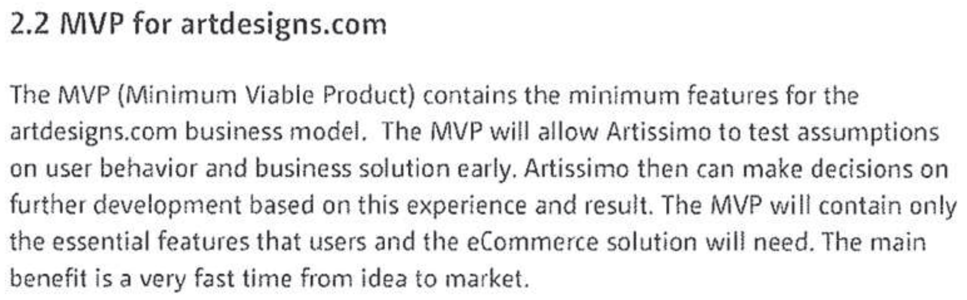


In the first phase Neofonie and Artissimo created a product backlog and specification for artdesigns.com. Based on this in the second project phase now a MVP (Minimum Viable Product) for artdesigns.com is planned to be developed and launched in September 2016. This MVP shall contain the minimum feature set for the artdesigns.com business modell in order to ensure the fastest time to market possible.

19. As can be seen in the excerpt below, this document indicates that both parties agree that MVP should "contain the minimum features for the artdesigns.com business model" and "the MVP

will contain only the essential features that users and the e-commerce solution will need." Also, the document states, "The main benefit is a very fast time from idea to market." [3]

Figure 2. Purpose of creating an MVP according to Phase 2 Offer



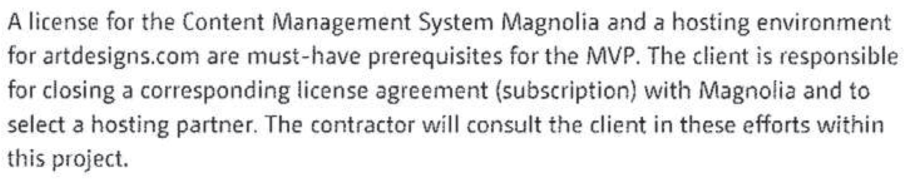
2.2 MVP for artdesigns.com

The MVP (Minimum Viable Product) contains the minimum features for the artdesigns.com business model. The MVP will allow Artissimo to test assumptions on user behavior and business solution early. Artissimo then can make decisions on further development based on this experience and result. The MVP will contain only the essential features that users and the eCommerce solution will need. The main benefit is a very fast time from idea to market.

20. I also reviewed Section 2.7 of Phase 2 Offer that describes the MVP for the Website as defined at the start of the project. The MVP includes a set of deliverables that are mapped to certain features and sub-features.

21. The Phase 2 Offer document also clearly identifies "acquiring a license to Content Management System Magnolia among the "must-have prerequisites for the MVP".

Figure 3. MVP must-have prerequisites

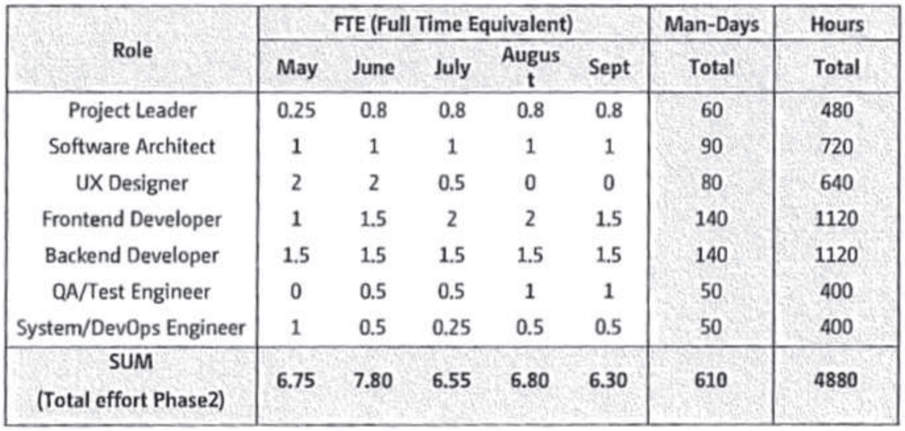


A license for the Content Management System Magnolia and a hosting environment for artdesigns.com are must-have prerequisites for the MVP. The client is responsible for closing a corresponding license agreement (subscription) with Magnolia and to select a hosting partner. The contractor will consult the client in these efforts within this project.

22. Furthermore, according to Phase 2 Offer both parties contemplated this phase to take about 5 months targeting Oct 10, 2016 as "End of Project".[4] The two sides stipulated phase 2 to take about 610 Man-Days or 4880 hours.

[3] Phase 2 Offer, page 5.
[4] Phase 2 Offer, page 7.

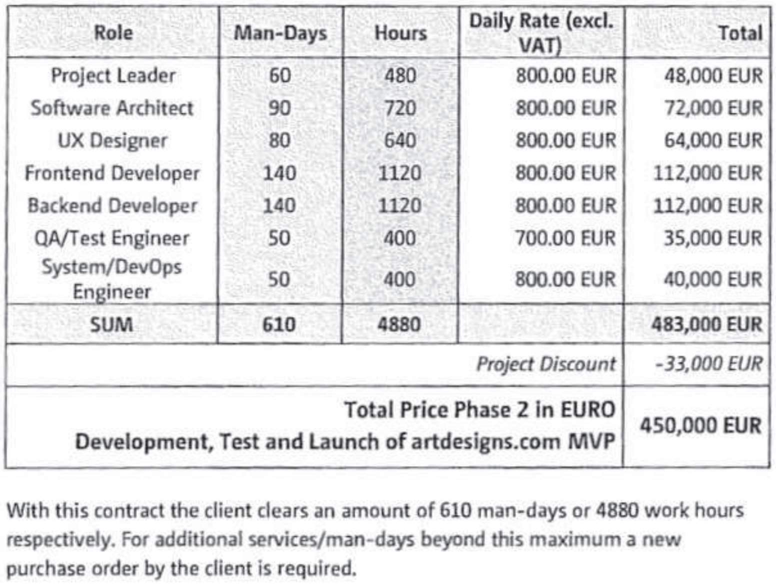
Figure 4. Estimated efforts for phase 2



Role	FTE (Full Time Equivalent)					Man-Days	Hours
	May	June	July	August	Sept	Total	Total
Project Leader	0.25	0.8	0.8	0.8	0.8	60	480
Software Architect	1	1	1	1	1	90	720
UX Designer	2	2	0.5	0	0	80	640
Frontend Developer	1	1.5	2	2	1.5	140	1120
Backend Developer	1.5	1.5	1.5	1.5	1.5	140	1120
QA/Test Engineer	0	0.5	0.5	1	1	50	400
System/DevOps Engineer	1	0.5	0.25	0.5	0.5	50	400
SUM (Total effort Phase2)	**6.75**	**7.80**	**6.55**	**6.80**	**6.30**	**610**	**4880**

23. Phase 2 Offer document also includes a detailed quotation on the total price for phase 2 based on the quoted rates for various roles involved in the project.

Figure 5. Phase 2 Offer quotation

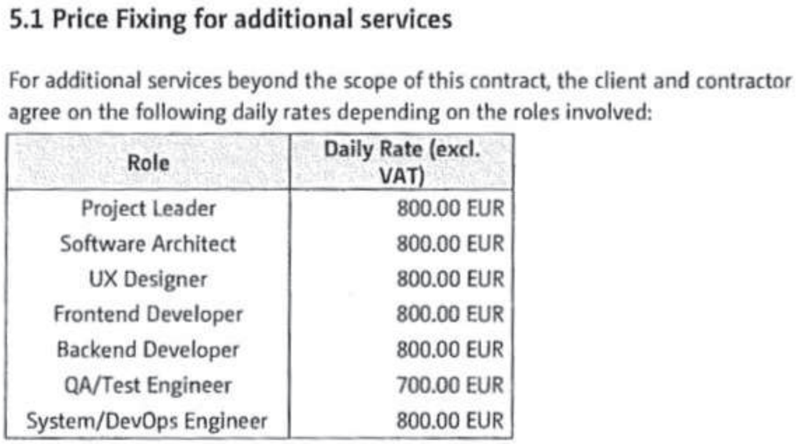


Role	Man-Days	Hours	Daily Rate (excl. VAT)	Total
Project Leader	60	480	800.00 EUR	48,000 EUR
Software Architect	90	720	800.00 EUR	72,000 EUR
UX Designer	80	640	800.00 EUR	64,000 EUR
Frontend Developer	140	1120	800.00 EUR	112,000 EUR
Backend Developer	140	1120	800.00 EUR	112,000 EUR
QA/Test Engineer	50	400	700.00 EUR	35,000 EUR
System/DevOps Engineer	50	400	800.00 EUR	40,000 EUR
SUM	**610**	**4880**		**483,000 EUR**
			Project Discount	*-33,000 EUR*
			Total Price Phase 2 in EURO Development, Test and Launch of artdesigns.com MVP	**450,000 EUR**

With this contract the client clears an amount of 610 man-days or 4880 work hours respectively. For additional services/man-days beyond this maximum a new purchase order by the client is required.

24. The agreement also provides fixed daily rate pricing "for additional services beyond the scope of this contract."

Figure 6. Price fixing for additional services



5.1 Price Fixing for additional services

For additional services beyond the scope of this contract, the client and contractor agree on the following daily rates depending on the roles involved:

Role	Daily Rate (excl. VAT)
Project Leader	800.00 EUR
Software Architect	800.00 EUR
UX Designer	800.00 EUR
Frontend Developer	800.00 EUR
Backend Developer	800.00 EUR
QA/Test Engineer	700.00 EUR
System/DevOps Engineer	800.00 EUR

25. As can be seen in the Section 2.7 of the Phase 2 Offer, the parties agreed for Plaintiff to deliver the following items as part of phase 2 project completion.

- Source code and documentation,
- Infrastructure necessary for development and launch of an MVP version of the Website,
- Development of an MVP for the Website according to MVP definition,
- Launching the MVP of the Website,
- A proposed plan for future improvements and operation of the Website after launch.

3.4 Phase 3. Improvements of MVP

26. Section 2.5 of Phase 2 Offer document contemplates planning for phase 3 to improve the MVP and operations of the Website after the successful launch of the MVP.

4 TECHNOLOGY BACKGROUND

27. Developing complex software projects requires significant amount of planning, extensive use of best practice and well-known software methodologies, and finally a well-defined process for identifying and fixing software bugs, improving or enhancing existing feature sets and making changes to various components of the developed software product.

28. In this section, I provide a brief background on general principles widely used in the design and development of complex software products. First, I cover Agile software development methodology given Plaintiff's extensive use of this methodology. Then, I provide an overview of the choices made during the software development regarding the development of custom code vs. utilization of third-party ("Off-The-Shelf") components. I also elaborate on an important software project management principle called The Project Management Triangle (and the notion of Minimum Viable Products as part of that discussion). Furthermore, I provide a review of key principles taken into account in the course of developing a Minimum Viable Product (MVP). This section provides the basis and context for my observations and opinions stated in the rest of the report.

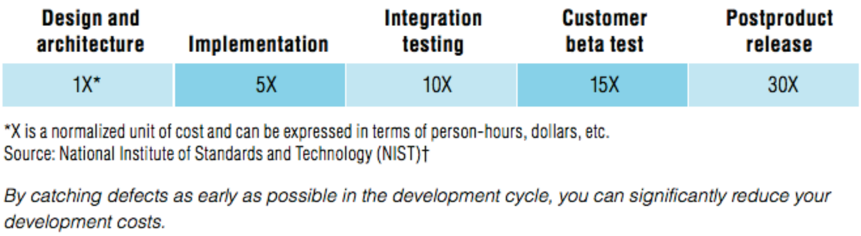
4.1 Agile Software Development

29. Traditionally, software development followed what is known as the Waterfall Model development methodology. Under this model, first software product requirements are first identified. Then, during the design phase decisions are made about the software architecture as well as its key components before starting the third phase, which is software development. Once the software development phase is complete, software testing begins to rectify bugs and other issues identified during the development.

30. It is clear from the above description that such models of software development were very linear in nature. Each phase would only begin after previous phase was completed. While seemingly simple in nature, employing such techniques for large and complex software projects created several issues. This is primarily due to the fact that the time and effort it takes to fix a software error significantly increases the longer it takes to identify it.

31. The white paper titled "Development solutions White Paper" published in October of 2008 is one of the numerous publications that quantifies the exponential increase in cost of fixing defects in later phases of the software development lifecycle. The report states that "costs of discovering defects after release are significant: up to 30 times more than if you catch them in the design and architectural phase."[5] The same report also states that "A large percentage of software development costs are spent on identifying and correcting defects."[6]

Figure 7. Exponential cost of fixing bugs late in software development life-cycle[7]



Design and architecture	Implementation	Integration testing	Customer beta test	Postproduct release
1X*	5X	10X	15X	30X

*X is a normalized unit of cost and can be expressed in terms of person-hours, dollars, etc.
Source: National Institute of Standards and Technology (NIST)†

By catching defects as early as possible in the development cycle, you can significantly reduce your development costs.

32. In another example, Steve McConnel in his book Code Complete, states that:

"Studies over the last 25 years have proven conclusively that it pays to do things right the first time. Unnecessary changes are expensive. Researchers at Hewlett-Packard, IBM, Hughes Aircraft, TRW, and other organizations have found that purging an error by the beginning of construction allows rework to be done 10 to 100 times less expensively than when it's done in the last part of the process, during system test or after release." [8]

[5] Minimizing code defects to improve software quality and lower development costs, IBM Rational Software Analyzer and IBM Rational PurifyPlus software, October 2008, page 1.

[6] Minimizing code defects to improve software quality and lower development costs, IBM Rational Software Analyzer and IBM Rational PurifyPlus software, October 2008, page 3.

[7] Minimizing code defects to improve software quality and lower development costs, IBM Rational Software Analyzer and IBM Rational PurifyPlus software, October 2008, page 2.

[8] Code Complete: A Practical Handbook of Software Construction, 2nd Edition (Paperback), page 29.

Figure 8. Increase in cost of fixing defects as time progresses[9]

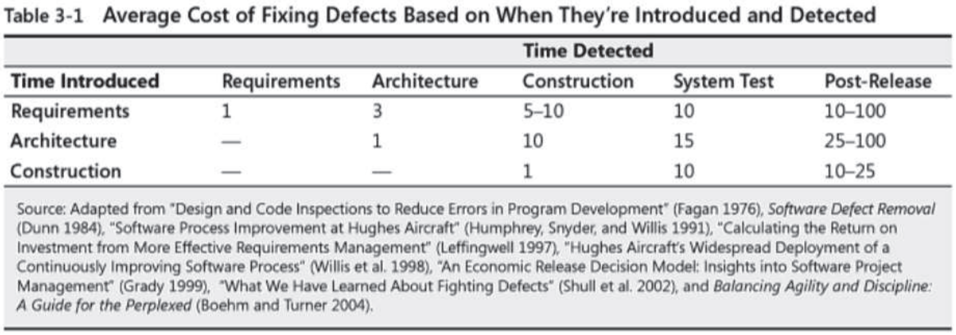


Table 3-1 **Average Cost of Fixing Defects Based on When They're Introduced and Detected**

	Time Detected				
Time Introduced	Requirements	Architecture	Construction	System Test	Post-Release
Requirements	1	3	5–10	10	10–100
Architecture	—	1	10	15	25–100
Construction	—	—	1	10	10–25

Source: Adapted from "Design and Code Inspections to Reduce Errors in Program Development" (Fagan 1976), *Software Defect Removal* (Dunn 1984), "Software Process Improvement at Hughes Aircraft" (Humphrey, Snyder, and Willis 1991), "Calculating the Return on Investment from More Effective Requirements Management" (Leffingwell 1997), "Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process" (Willis et al. 1998), "An Economic Release Decision Model: Insights into Software Project Management" (Grady 1999), "What We Have Learned About Fighting Defects" (Shull et al. 2002), and *Balancing Agility and Discipline: A Guide for the Perplexed* (Boehm and Turner 2004).

33. Regardless of exact amount of increase in complexity and resources it takes to fix software defects, it is important to make two important observations. First, proper use of software engineering principles causes a dramatic reduction in the overall cost of software development. Secondly, delays in identifying software defects or changes in existing functionality can massively increase the cost to rectify them (e.g., fixing a defect, modifying an existing feature) the later they are introduced.

Figure 9. Relative cost to fix defects in each phase of software development[10]

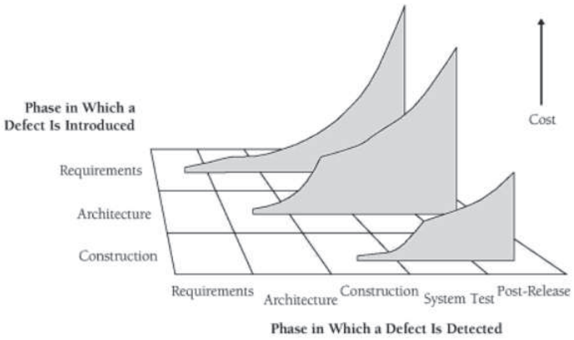


Figure 3-1 The cost to fix a defect rises dramatically as the time from when it's introduced to when it's detected increases. This remains true whether the project is highly sequential (doing 100 percent of requirements and design up front) or highly iterative (doing 5 percent of requirements and design up front).

[9] Code Complete: A Practical Handbook of Software Construction, 2nd Edition (Paperback), page 29.
[10] Code Complete: A Practical Handbook of Software Construction, 2nd Edition (Paperback), page 30.

34. A significant amount of work in software engineering is dedicated to devising frameworks and processes that reduce such overall cost by flattening the cost of changing software. One of the most popular approaches to achieve this goal is to develop software in increments in what is knowns as the Spiral Model (in contrast to the Waterfall Model).

35. Spiral software development methodologies promote breaking the overall software development process into smaller iterations (also known as spirals or cycles). For each iteration, a set of requirements are identified during a planning phase and risks or alternate solutions are identified for the captured requirements. Software components are then developed to implement the requirements identified during that iteration and are subsequently tested. Finally, during the evaluation phase, software developed is reviewed and evaluated to plan for the next iteration. This process is repeatedly performed in iterations where each subsequent iteration builds on what has been accomplished in previous iterations.

36. Despite the fundamentally different characteristics of the Spiral and traditional Waterfall software development methodologies, neither approach offers a “perfect” project requirement phase that can comprehensively capture customer needs without requiring further modifications.

37. On one end of the spectrum, the Waterfall model requires a detailed and rigid requirement specification phase, which has the advantage of spending more time on identifying requirements. However, it is almost always the case that requirements change over time (often quite significantly) and the cost of change in requirements is dramatically increased in later stages of software development as discussed above. The other end of spectrum consists of Spiral methodologies that have a much less strict emphasis on capturing a comprehensive set of requirements up front. These methodologies greatly increase flexibility and reduce cost, while by definition, causing more requirements to be discovered (or existing features to be modified) along the way.

38. Agile software development further builds on the iterative nature of the Spiral software models while also focusing on a key principle: embracing change. There are built-in mechanisms in Agile development that allow a change in scope by emphasizing concurrency in software development activities throughout the project. In contrast to Waterfall Models (where a phase begins after previous phase ends), development activities do not end and continue throughout the project. The concurrent nature of development tasks and the breakdown of software development into smaller cycles give Agile more flexibility and openness to constantly changing requirements and provide means for timely capturing user feedback, prioritization of tasks and triage of defects. Agile promotes early and continuous delivery of software, and regular intervals for meeting between stakeholders to reflect back and fine-tune the process to be more effective in the future.[11]

39. Like any other process, Agile has its own disadvantages as well. Given its flexibility to welcome continuous feedback and its reduced emphasis on fully developed and well-defined requirements specification in the early phases of a project, the final software product can be significantly different from what was originally envisioned. However, welcoming change based on client feedback can very well create additional value beyond the original vision for a product. Lastly, Agile encourages measuring the amount of working software as an indication of progress instead of heavy documentation. While comprehensive documentation does not necessarily lead to successful projects, it is important for the development team to ensure proper level of documentation throughout the software development lifecycle.

4.1.1 Kanban

40. There are several well-known methodologies that are used to implement Agile thinking in the software development process by sharing its key principles. I studied the Phase 2 Offer

[11] Lean Software Development: An Agile Toolkit, Mary Poppendieck, Tom Poppendieck. Publisher: Addison Wesley Pub Date: May 08, 2003.

document which indicates Plaintiff proposes using Kanban methodology for the Website.[12] Therefore, in the following sections I provide more context and background around important constructs of Kanban methodology as they are related to this case.

41. Kanban is a Japanese term which loosely translates to “sign,” “card,”, or “billboard.” Kanban was originally developed at Toyota to improve the flow of parts in their production lines in the 1950s. Kanban is used by several world-renowned companies such as BBC[13], SAP[14], Siemens[15] and many more for their software development endeavors.

42. By introducing visual elements into the process, such as a Kanban board, Kanban was later adopted to be used as an Agile software development methodology. Kanban encourages breaking the task of developing complex software into smaller tasks that result in incremental changes to the project and pulling work from a prioritized set of tasks.

43. A Kanban board offers a visual way to represent the status of each task and the overall flow of the project for all team members. Kanban boards typically include a multi-step workflow such as “To-Do”, “In Progress” and “Complete”. Teams employing Kanban typically create additional columns for the board to visualize and track a more detailed workflow. My analysis of Neofonie’s implementation showed that the development team used an 8-step workflow that included the following phases: “Dev Analysis,” “Dev Ongoing,” “Review,” “Dev Done,” “QA Ongoing,” “QA Done,” “Ready for Deployment,” and “LTS”.[16]

[12] Phase 2 Offer, Section 2.3.
[13] Lean Software Management: BBC Worldwide Case Study, Peter Middleton and David Joyce, IEEE Transactions on Engineering Management
[14] Two years of applying Kanban at SAP: a report from the trenches (Alexander Gerber & Martin Engel). https://vimeo.com/80897039 and accompanying presentation.
[15] Kanban at Scale – A Siemens Success Story https://www.infoq.com/articles/kanban-siemens-health-services
[16] Some of the content I reviewed in the board is in German. I used Google translate to translate them back to English.

44. In contrast to the linear nature of the Waterfall software development, Agile methodologies such as Kanban promote a continuous cycle. In each cycle, first a set of requirements are captured (if needed by including managers, key stakeholders and users). The requirements are not intended to capture all functionality of features of the product; rather, the requirements only address a limited subset of the functionality or features. These requirements are then mapped to a set of features during a planning phase, user stories are created that capture key end-user needs and wireframes are created to provide an abstract and high-level view of the user experience (UX) as they interact with various parts of the product.

45. The planning phase is followed by a design phase. In the design phase, key system components are identified and the overall system architecture that allows these components to interoperate is created. For the visual aspects of the project, wireframes are enhanced and turned into detailed design sketches that can at times be "pixel perfect" illustrating exactly how the final product would look like. The development follows the design stage and once a task is complete, it is released (typically deployed to a live system) to be reviewed by the end-customer. Once the customer feedback is gathered, the cycle is repeated.

46. In order to implement Agile methodologies, software teams typically employ software development tools that allow them to drive a project and implement key principles of Agile. Atlassian's JIRA[17] is among the most popular software development tools used for this purpose. JIRA enables development teams to prioritize work, visualize work flow, assign tasks, monitor progress and comment on each task. JIRA is being widely used in the software industry by numerous organizations such as Audi, Autodesk, BAE Systems, Docker, indeed, NASA, Twitter and more.[18]

[17] https://www.atlassian.com/software/jira
[18] Atlassian Customers https://www.atlassian.com/customers

47. Entries in JIRA are typically called tickets and represent a task. The task can describe a new feature, explained from the perspective of the end user (and hence called a user story). JIRA tasks are also created to report a bug identified in the software. JIRA also provides a mechanism to describe relationship between various tasks. For instance, one can mark a task(s) as a pre-requisite to another task(s), or to split a task into a set of sub-tasks. Furthermore, “epics” are used to group related user stories or to break up a large user story into smaller chunks.

48. Tasks can be assigned to different users throughout the course of their existence. For instance, a reporter might find a bug and create a ticket for it accordingly. The ticket is then assigned to a developer to be addressed. Once the bug is fixed, the developer might assign the task to a QA team member to further confirm the bug has been removed. Once QA is complete, the task could be marked as “finished” and be re-assigned back to the reporter (or a different individual) who will subsequently mark the ticket as “closed.”

49. In summary, using Agile development methodologies such as Kanban, software systems are incrementally built by mapping customer requirements to a set of features, and in turn mapping those features to a set of tasks and iteratively and continuously going through pushing tasks through the software development lifecycle. Such methodologies provide visibility and accountability while enabling continuous deployment, prioritization, hypothesis validation and tracking of tasks through the use of a ticketing system and a Kanban board.

50. While Agile methodologies can significantly reduce the overall cost of software development, it is important to note that existence and extensive use of feedback loops play an equally significant role. Getting feedback from people using the system is one of the most effective ways of ensuring the software performs its stated goals.

51. To address this important requirement, Agile includes several constructs to gather user feedback at various phases of software development. For instance, identifying personas and user stories, design mock ups, prototypes, regular product demonstrations, recurring progress reports and UAT (user acceptance test) are all examples of such techniques. The iterative nature of Agile lends itself well to capturing timely user feedback at the end of each cycle and applying the necessary course-correction measures before it becomes prohibitively expensive or impossible to make the necessary changes. This makes user feedback an integral part of Agile and gives it a key advantage over more traditional (and linear) methods of software development. No amount of additional time and resources spent by the software development team in testing the product can equate or replace timely feedback received from the customer who is the main entity with domain expertise and a true understanding of end-user.[19]

4.2 The Project Management Triangle

52. Despite Agile's native support for change, constant shift or increase in project scope remains a major threat to overall success of any project in general and any software project in particular. To illustrate this important concept and in order to better understand the effect of scope change in a (software) project, I now discuss an important topic in project management known as The Project Management Triangle.

53. Every project is constrained by a set of factors that govern how it is being perceived, planned and executed with a direct effect on its outcome. In the world of software, there is a routinely used principle referred to as the Project Management Triangle (PMT) that provides an important framework and a strong foundation to better understand and manage complex software

[19] Lean Software Development: An Agile Toolkit, Mary Poppendieck, Tom Poppendieck. Publisher: Addison Wesley Pub Date: May 08, 2003.

projects. The Project Management Triangle[20] is founded on a key principle: product quality is constrained by three key attributes: *budget*, *timeline* and *scope*.[21] The Project Management Institute provides a more formal definition of these three constraints:

> *"Cost is a function of time and scope, that these three factors are related in a defined and predictable way"*[22]

54. I now briefly define these three key principles. Budget is an indication of the overall cost to complete the project. Timeline (also referred to as schedule) determines the amount of time it takes to complete a project. Lastly, Scope (also referred to as features) indicates the deliverables for the project such as various functionality that the product is going to include.

55. The basic idea behind PMT is that there is a competing nature between the three above constraints in any (software) project. Gravitating toward any of the constraints will inadvertently have a negative effect on the other two. For instance, increasing scope would cause projects to take longer to finish and/or cost more to complete (if same quality is expected). Similarly, reducing the overall cost of the project means either the scope needs to be reduced or time to complete the project should be expanded.

56. A critical teaching of the above model is that the development team should always strike a balance between these competing factors to reach an acceptable trade-off between them. This model also shows why increased attention to any of the above factors should be compensated by the other two in order for overall project quality not to suffer.

[20] The Project Management Principle is sometimes also referred to as The Iron Triangle, The Triple Constraint, PM's Pyramid or the Time Cost Quality Triangle.

[21] A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Project Management Institute. 2000 Edition.

[22] Project Management Institute, https://www.pmi.org/learning/library/triple-constraints-success-additional-factors-6591

4.3 Minimum Viable Product

57. One of the biggest challenges in developing products is ensuring they address the end customer need while minimizing engineering and design resources. The lean startup movement, popularized by Eric Ries, promotes the idea of developing what is called a "Minimum Viable Product" or MVP for short. MVPs are primarily created and used as one of the most effective ways of reducing time to market by going through a series of "build-measure-learn feedback loops."[23]

4.3.1 Build, Measure, Learn Feedback Loop

58. The notion of Minimum Viable Product is more formally defined as follows

> *"A Minimum Viable Product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort."*[24]

59. The basic idea behind the above definition is identifying the leanest product one can build that delivers value to the end customer while enabling the product owner to quickly build the software, measure how it is perceived by the end user through validating a set of hypotheses and learn from customer feedback before going through the cycle again. Therefore, minimizing the path to get customer feedback, even if it means some features are not fully developed is a key principle in the development of an MVP. This notion is best captured by the following quote from Reid Hoffman, the founder of LinkedIn:

> *"If you are not embarrassed by the first version of your product, you've launched too late."*[25]

[23] The Lean Startup, Chapter 6. Eric Ries, Portfolio Penguin. 2011.
[24] Minimum Viable Product: a guide. http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html
[25] What LinkedIn Founder Reid Hoffman Learned From an Early Failure, https://www.wsj.com/articles/what-linkedin-founder-reid-hoffman-learned-from-an-early-failure-1525140240

60. Seasoned software development teams always strive to find the right balance between shipping a product too early vs. too late. This is because delaying the launch to "perfect" certain features prevents a timely validation of a hypothesis and instead leaves the stakeholders with a product built based on unproven assumptions. Such a product is very likely to miss key features and instead include useless features no one would use:

> *"The approach we took was to complete the entire product before we pulled back the curtain and let people sign up. This approach delayed SocialNet's launch by a year, and when we finally did launch, we quickly realized that half of the features we'd painstakingly implemented weren't important, and half the important things that our service would be useless without were missing, because we hadn't thought of them."*[26]

61. One basic test to determine if a feature should be included in the MVP is to see if a product can be shipped and used by the end-users without that feature. For instance, when building an e-commerce application, being able to browse through product offerings, or being able to add a product to a shopping cart qualify as MVP features as the product cannot simply function without them. In contrast, being able to track a shipped product or receive an email alert when the price of an item drops would not be considered MVP features since the product clearly functions without their presence. One might note however that the price drop alert feature might very well be considered an MVP feature for a product specifically designed to alert users when lower pricing is available for an item. For such a product, however, existence of a shopping cart user flow might not be considered an MVP feature.

[26] What LinkedIn Founder Reid Hoffman Learned From an Early Failure, https://www.wsj.com/articles/what-linkedin-founder-reid-hoffman-learned-from-an-early-failure-1525140240

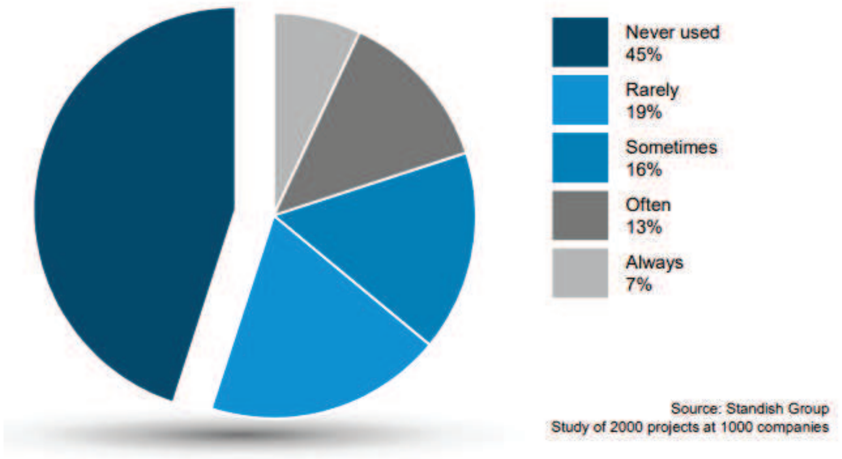
62. Building an MVP allows companies to reduce the time to market (and in turn development costs) while being able to quickly capture end user feedback. Some of the world's most successful software companies created an MVP as an early version of their product. These MVPs include early prototypes at places such as Twitter, Dropbox, Groupon, Zappos, and many more.[27]

63. In the course of envisioning, designing and developing software projects, there are several hypotheses about end user's expectation and needs for the product being built. The challenge being faced in almost any major software development project, however, is how to validate such hypothesis as quickly and as cost-effectively as possible. The development of MVPs for early testing of hypotheses has become a widely popular practice in the software development industry for this very reason. MVPs allow teams to create an environment when products can go to market fast, feedback can be gathered early, and hypotheses can be validated before investing a significant amount of upfront time and energy to design features customers won't need or use. MVPs are used to battle the fact that a vast majority of software projects contain a significant set of features that very few (or no) users use.

64. In summary, MVP is not just a product, it is a different way of thinking about building products. It is a methodology that prevents teams from going through constant changes in scope *before* the product even launches. Instead it promotes releasing the product as early as possible and going through iterations to refine things only *after* the product has been used by its end customers.

[27] 15 Ways to test your minimum viable product https://thenextweb.com/dd/2014/11/12/15-ways-test-minimum-viable-product/

Figure 10. Feature usage breakdown of typical software systems[28]



Source: Standish Group
Study of 2000 projects at 1000 companies

4.4 Integration with Out-of-the-Box Software

65. In the course of the design and development of software products, the engineering team must always choose between implementing various components in-house and building them from scratch vs. using third-party, Out-of-the-Box (OOTB) software components.[29] For many projects, using OOTB packages can greatly increase productivity, reduce costs and shorten time to market. There are also numerous cases where (extensive) use of OOTB software can lead to significant complexity, long delays and massive cost increases for the development of a software product (in an extreme case, it might even lead to complete project failure). There are many factors that must be taken into account when making the decision for whether OOTB software should be used or not.

4.4.1 When to Use OOTB Software

66. In the world of software development, there are many instances where opportunistic use of OOTB software is recommended as a good design and development practice. Especially when the

[28] The Pay TV Market is Changing. https://www.cisco.com/c/m/en_us/network-intelligence/service-provider/digital-transformation/vodafone-launches-giga-tv.html

[29] These already developed components are also called Commercially-Off-The-Shelf or COTS packages.

overall scope of a project is fairly simple, requirements are known in advance, very little to no customization is needed and "standing out" from the crowd is not required, employing OOTB packages can become very beneficial.

67. Under the right circumstance, employing OOTB solutions reduces upfront costs while typically providing a more feature-rich alternative to custom development. (This is because such packages are developed for a broad audience and are intended for extended reuse.) In addition, wide adoption of these packages can generally make them more robust since bugs have been already discovered and fixed based on feedback provided by earlier adopters of the packages. In other words, the time and resources of designing, developing and testing a certain component or package has been already handled by the entity who has developed that component or OOTB package. However, the task of integrating the OOTB package with the rest of the software and performing further integration tests still relies on the development team using the package.

68. Another benefit of using OOTB packages is expected behavior. Since the packages are typically tried and tested before, there is less chance of surprises down the road since behavior has been verified or at least observed before the decision is made to integrate it into the overall system.

4.4.2 When Not to Use OOTB Software

69. As the number of software components needed for a product, and their complexity increase, it becomes increasingly infeasible to use OOTB packages. In this section, I discuss some of the reasons that render excessive use of OOTB packages a poor choice for such software projects.

70. When a software project consists of several components, there will be a need for these components to communicate with each other. A widely used principle in software development to facilitate such interactions is for each software component to expose its key functionality through a set of *interfaces* while encapsulating its inner workings from the outside world. Any other part

of the system that needs to interact or employ a component needs to use its well-defined public interface and its expected communication protocol. Therefore, when integrating an OOTB package, the developers must write custom code to ensure interoperability between that package and the rest of the system.

71. As the number of OOTB packages being used in a software system increases, integration becomes increasingly more complex and at times utterly infeasible. This is primarily due to the fact that the overall system becomes increasingly more rigid and less easy to adjust and maintain. Moreover, consider the case where OOTB package **A** needs to communicate with OOTB package **B** but their interfaces are not compatible. Since by design, there is usually no easy way of modifying either package, the developer needs to write additional custom code to facilitate this communication. Given the high amount of recurring challenges faced by developers when trying to reuse previously developed code, a well-known and commonly used software design pattern called the Adapter Pattern (also known as the Wrapper Pattern) has become popular in reducing integration costs and to increase interoperability.[30]

72. Encapsulating inner workings of a software component from the outside world also means that there is usually no feasible way of changing the functionality and appearance of an OOTB package beyond its limited set of exposed preferences. Therefore, if the default appearance or behavior of an OOTB package is not desired, or if any customization behavior or appearance is needed, use of OOTB software is not recommended.

73. OOTB packages are very unlikely to meet all business needs. On the contrary, they typically include features that are not desired or needed by the client. More specifically for visual

[30] Adapter design pattern, w3sDesign http://w3sdesign.com/?gr=s01&ugr=struct

components, even if all desired features exist, the look and feel might not fully conform to the overall design of the software product.

74. OOTB packages might require certain infrastructure and platform to run properly that the business cannot (at least without spending significant resources) provide.

75. Perhaps the most important disadvantage of using OOTB solutions is maintenance (and in turn obsolescence). OOTB packages might not get upgraded as new technologies, standards, regulations, compliance or security vulnerabilities are introduced or support for them might dwindle over time.

76. Lastly, future feature requests would have to be almost completely disregarded in the presence of many OOTB packages since by design, they are not meant to be customized or enhanced over time. Therefore, in cases where learning from user feedback is going to shape future product features, use of OOTB can introduce additional problems which might ultimately led to a complete abandonment of them and replacing them with custom developed code.

77. The trade-offs stated above have been widely discussed in the software development community and various guidelines are usually suggested to help development teams identify and find the right approach given their project requirements.[31]

4.5 Analysis Methods

78. This section explains the analysis methods I employed in the course of forming the opinions expressed in this report.

[31] Build vs. Buy: How to Know When You Should Build Custom Software Over Canned Solutions. https://www.forbes.com/sites/chuckcohn/2014/09/15/build-vs-buy-how-to-know-when-you-should-build-custom-software-over-canned-solutions/

4.5.1 Software Architecture Recovery

79. In the course of forming my opinions, I applied an established analysis method known as software architecture recovery. Software architecture recovery comprises a family of related and complementary techniques for discovering and documenting the principal design decisions about a software system from its implementation-level artifacts.[32] The principal design decisions are those design decisions that are most important and dictate the fundamental structure and behavior of the system.

80. The goal of a software architecture recovery effort is to document the architecture of a complex software system in such a way that questions relating to design and operation of the system may be answered. Among other things, software architecture recovery seeks to identify:

- the nature of interactions and relationships between software components;
- the rules and constraints governing the assembly and integration of components;
- the beneficial structural and behavioral patterns embodied in the software; and
- the rationale underlying the system's design.

81. Software architecture recovery is performed by gathering information from the system's source code, available technical documentation, and stakeholder interviews (such as depositions). There are different ways and methods of performing software architecture recovery such as automatic, semi-automatic and manual methods. However, a manual recovery that is performed by someone trained software expert, although being labor-intensive, is the most reliable and authoritative method for software architecture recovery.[33] I used my experience in development

[32] J. Garcia, I. Krka, C. Mattmann, and N. Medvidovic, "Obtaining Ground-truth Software Architectures," in Proc. ICSE, ser. ICSE '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 901–910.

[33] Joshua Garcia, Ivo Krka, Nenad Medvidovic, and Chris Douglas. "A Framework for Obtaining the Ground-Truth in Architectural Recovery." Proceedings of the 2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), Aug. 2012.; see also Jingwei Wu, Ahmed E. Hassan, and Richard C. Holt. "Comparison of clustering algorithms in the context of software evolution."

of numerous R&D and commercial technology-heavy software applications to determine overall structure by performing manual recovery in this case.

4.5.2 Software Process Assessment

82. Software process assessment is used to determine how a software product is designed and developed within the constraints of cost, schedule and quality. The assessment involves an appraisal/review of how software is developed by a team and comparing it, to the extent possible/applicable, to processes used in the industry for developing similar projects.

83. Software process assessment can be qualitative or quantitative. In either approach, the goal is to use a set of established methods such as inspecting the procedural steps, relating them to results obtained, information regarding the amount and severity of defects identified and the time it takes to address and resolve the defects to determine the quality of software process employed.

84. In preparing this report, I analyzed JIRA tickets, email exchanges and other relevant case documents outlined in Exhibit 2 to understand the process employed by Plaintiff in developing those features. I also analyzed such evidence to study changes over time in scope, structure and development of disputed deliverables. I traced changes, modifications and updates related to such deliverables beginning from Phase 2 Offer until the date Defendant decided to end its relationship with Plaintiff.

85. By connecting various pieces of evidence and activities surrounding such disputed deliverables for the Website over time, I recreated a more accurate and complete timeline of the process as well as the design, implementation and architectural decisions made by the Plaintiff in light of Defendant's actions (or lack thereof).

86. This process involved studying the scope of the deliverables reflected in Phase 2 Offer document, details such as time of creation and acceptance criteria captured in tickets created in

JIRA corresponding to those deliverables, activities by Defendant in form of comments left in JIRA for those tickets, email communications or notes captured during recurring meetings between the two sides, status updates by Plaintiff and analyzing activity logs for JIRA tickets such as changes in ticket type, priority, status, epics and solution version. [34]

87. Due to the unique nature of software projects, they are more susceptible to complaints by software developers about incomplete or constant changes in requirements and complaints by customers about how well the end product met the stated requirements.[35] Therefore, a careful analysis of evidence pieced together over time help shed light on the root cause and nature of budget overruns or delays in project completion.[36]

5 FINDINGS AND OPINIONS

88. This section enumerates my opinions based on the results of my analysis.

5.1 Summary of Analysis Results

89. This section contains the summary of my findings after performing my analysis of the documents produced in this matter. My analysis determined that:

90. Defendant failed to conform to important principles of Agile software development and MVP launch. In my opinion, this resulted in unnecessary delays in the development of the Website.

91. Plaintiff chose a widely practiced software development process and implemented it correctly in the course of developing the Website.

[34] In course of preparing this report, I did not analyze artifacts such as actual source code or a live version of the Website that would allow me to form an opinion about the quality of delivered features at the end of the relationship between the two parties. I am willing to amend or supplement this report if such material is made available to me.

[35] Introduction to Software Process Improvement, Watts S. Humphrey, Software Engineering Institute Carnegie Mellon University.

[36] Delivering large-scale IT projects on time, on budget, and on value. Michael Bloch et al., McKinsey & Company https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value

92. Plaintiff stroke a balance between integrating third-party and existing software components and developing them from scratch. In addition, Plaintiff's decision to integrate out-of-the-box software components when applicable and to avoid excessive use of such components were justified.

93. In its response to Plaintiff's complaint, Defendant did not accurately represent the severity of several bugs that were discovered in the course of developing the Website as well as Plaintiff's approach in addressing or resolving such bugs.

94. Plaintiff was unable to successfully complete the development of the Website in a timely manner due to Defendant's inability or unwillingness to take several necessary actions that would unblock Plaintiff and would enable it to successfully complete the development of the Website.

5.2 Defendant's Lack of Conformity to Principles of Agile Development of an MVP Caused Unnecessary Delays in Developing the Website

95. My analysis of communications between the two parties over time as well as JIRA and Confluence logs demonstrated that Defendant did not properly follow key principles of Agile thinking and MVP development which negatively affected the planned launch of the Website.

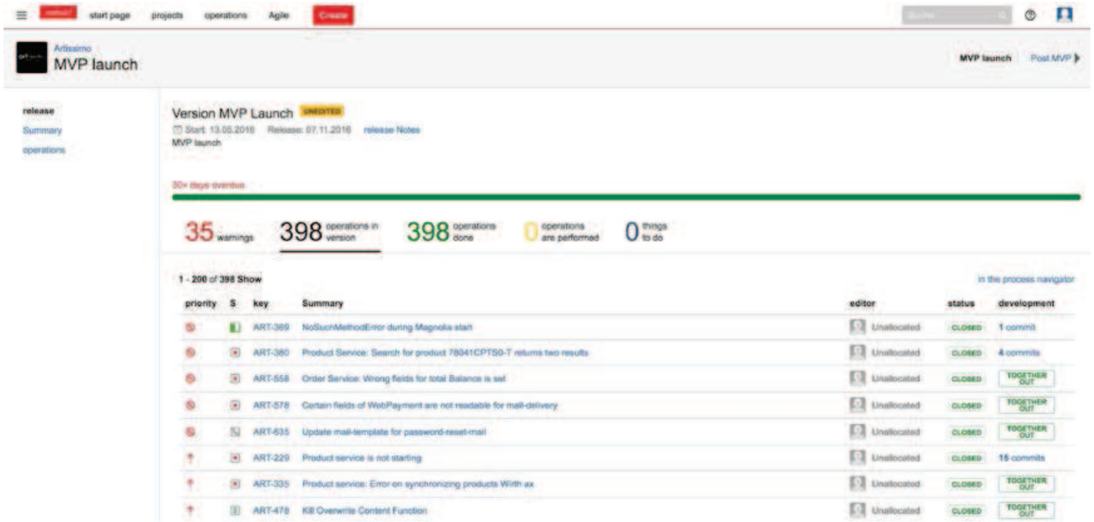
96. As discussed in Section 4.3, a Minimum Viable Product, by definition, is considered to contain the minimum set of features to be viable. I studied Section 2.7 "Deliverables for Phase 2" of the Phase 2 Offer Document which contains a high-level description of features the parties decided to include as part of the MVP. Section 2.7 includes a total of about 63 features and sub-features.

97. I also studied the tasks created in JIRA for various phases of the project such as "MVP Launch", "post MVP" and "Public Launch Blocker" and of various types such as "User Story"[37],

[37] User stories provide a non-technical description of what needs to be developed, from an end-user's perspective.

"Bug", "Sub-Task" and "Epic"[38]. Upon further analysis, I found a total of 398 tasks labeled "MVP Launch" representing the tasks related to the MVP product for Defendant.

Figure 11. Screenshot of tasks marked as MVP Launch



Version MVP Launch UNEDITED
Start: 13.05.2016 Release: 07.11.2016 release Notes
MVP launch

30+ days overdue

35 warnings 398 operations in version 398 operations done 0 operations are performed 0 things to do

1 - 200 of 398 Show — in the process navigator

priority	S	key	Summary	editor	status	development
		ART-369	NoSuchMethodError during Magnolia start	Unallocated	CLOSED	1 commit
		ART-380	Product Service: Search for product 78041CPTS0-T returns two results	Unallocated	CLOSED	4 commits
		ART-558	Order Service: Wrong fields for total Balance is set	Unallocated	CLOSED	TOGETHER OUT
		ART-578	Certain fields of WebPayment are not readable for mail-delivery	Unallocated	CLOSED	TOGETHER OUT
		ART-635	Update mail-template for password-reset-mail	Unallocated	CLOSED	TOGETHER OUT
		ART-229	Product service is not starting	Unallocated	CLOSED	15 commits
		ART-335	Product service: Error on synchronizing products Wirth ax	Unallocated	CLOSED	TOGETHER OUT
		ART-478	Kill Overwrite Content Function	Unallocated	CLOSED	TOGETHER OUT

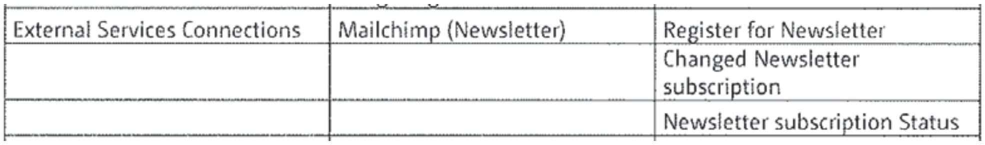
98. As documented in JIRA logs, Plaintiff has developed a detailed and meticulous process to keep track of numerous tasks related to the MVP launch. These hundreds of tasks are each linked to wiki pages relevant to the task (if applicable), and often contain attachments, screenshots, detailed comments and a comprehensive log of all activities related to each task.

99. As discussed in Section 4.5.2, in order to better understand the effect of changing the scope and level of detail of various features requested by Defendant, it is helpful to take a closer look at the disputed tasks and view them through the lens of time. By tracking the evolution of a product feature specification, from inception to implementation and later deployment, one can gain a better understanding of the root causes for the delay completing the MVP product.

[38] Epics are used to group related user stories or to break up a large user story into smaller chunks. See more discussion in Section 4.1.

100. One example of such change in a product scope is the design of a newsletter for the Website. Section 2.7 of the Phase 2 Offer specifies the following three sub-features for the newsletter:

Figure 12. Deliverables for newsletter (features and sub-features)



External Services Connections	Mailchimp (Newsletter)	Register for Newsletter
		Changed Newsletter subscription
		Newsletter subscription Status

101. As the above list suggests, the original scope of the newsletter feature was limited to allowing users to register for a newsletter, change their newsletter subscription, and to view their newsletter subscription status.

102. As discussed in Section 4.3, an important aspect of building a successful MVP using Agile development methodologies is to first identify a set of crucial features without which the product cannot be shipped, and then to implement a basic version of that identified feature as rapidly as possible to capture early user feedback, validate relevant hypotheses and decide on further development and enhancement of the feature.

103. The Phase 2 Offer specifically addresses these two key characteristics of an MVP for the Website in Section 2.2. where it states "The MVP (Minimum Viable Product) contains the minimum features for the artdesigns.com business model" and that "The main benefit is a very fast time from idea to market."

104. Unfortunately, the selection of those "minimum features" and their corresponding scope do not follow the two aforementioned MVP criteria: while it is impossible to launch a successful e-commerce website without a "shopping cart" or "checkout" feature, one could build an e-commerce website without a full-blown newsletter feature and still test a large number of

hypotheses about the end-user behavior, before investing significant time and resource perfecting the newsletter feature.

105. However, let's assume that Defendant firmly believed having a newsletter in the MVP was a critical component of its business model. I studied the requested changes to the scope of this feature throughout the project to analyze how well it captured the definition and scope of an MVP feature.

106. First, as suggested by Figure 12, Neofonie had to develop means for consumers to unsubscribe from an Artissimo newsletter. If real MVP thinking was present, one could simply use an "unsubscribe" button at the footer of the email being sent out as a means of unsubscribing from a newsletter. Instead, what was scoped out constituted creating a way of unsubscribing through the Website. This latter bloated approach entails dependence on other features such as authentication, requires creation of additional pages such as a page to manage newsletter subscriptions, requires programmatic integration with a third-party newsletter provider such as Mailchimp, and many more unnecessary actions.

107. On 06/21/2016, Defendant states that it is happy with the implementation of Mailchimp at that point by leaving a comment on the ticket named "Newsletter Component".[39]

108. Later, in a separate ticket created on 07/19/2016 for the "My Collection: Overview to all collections" feature, it is stated that "user can see the newsletter module and thereby sign up to the newsletter", asking for further integration of newsletter sign up feature into an already complex feature.[40]

109. Later, in a separate ticket created on 09/18/2016 for "My Account: Overview", it is stated that "As a user I want to see my order, user data, newsletter settings, billing and shipping

[39] https://jira.neofonie.de/browse/ART-125
[40] https://jira.neofonie.de/browse/ART-206

address in one view".[41] Given the disconnected nature of order data, user data, newsletter settings and addresses, one can see additional integration to a plurality of disparate services being needed to correctly implement this feature.

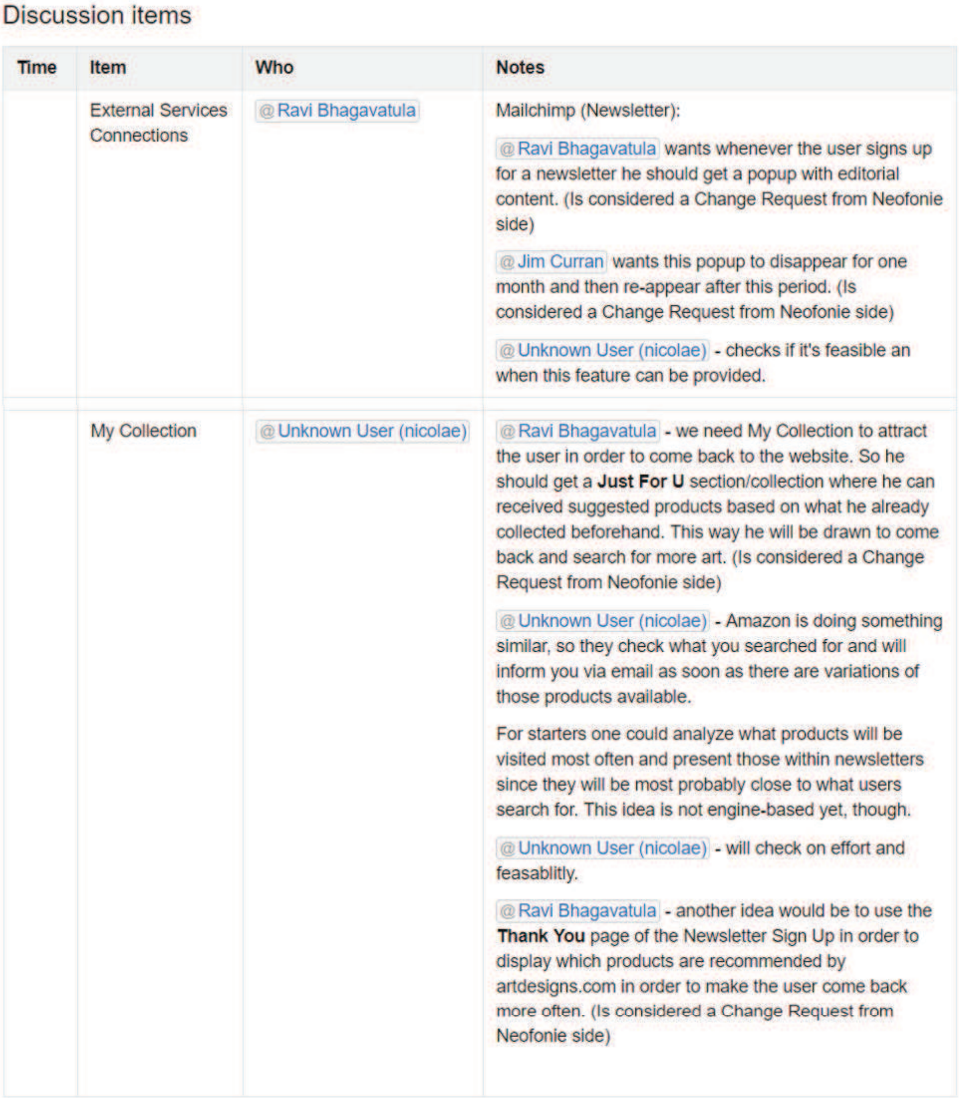
110. Later, in a separate ticket created on 10/11/2016 for "Newsletter Signup: Redirect Rule", it is stated that "As an editor I want to be able to set up content on the page for the users who just signed up to the newsletter. Since the user is being forwarded to the Mailchimp Thank You page this page has to be redirected to the page the editor set up."[42] Implementation of this feature requires the development of an additional custom page set up by the editors, implementation of a redirect logic that redirects user browsers from a generic "Thank You" page to a custom created one, and the provision of allowing editors to provide content for the custom "Thank You" page.

111. Later, a separate ticket created on 10/12/2016 for "Pop Up for Newsletter Promotion," provides additional requests for newsletter functionality by stating that "As a business owner I want to make a pop up with editable promotions (newsletter context) in order to get the user back to artdesigns.com."[43]

112. Later, in a confluence page titled "2016-10-10 Review Deliverables Part 2", there are notes captured from a meeting that has seemingly taken place on Oct 10, 2016. The meeting minutes list various attendees from both parties and states "Show artissimo the current status of all deliverables of the contract" as its stated goal.[44] I have captured two relevant sections from the meeting minutes of that day in the image below.

[41] https://jira.neofonie.de/browse/ART-403
[42] https://jira.neofonie.de/browse/ART-489
[43] https://jira.neofonie.de/browse/ART-498
[44] https://confluence.neofonie.de/display/AR/2016-10-10+Review+Deliverables+Part+2

Figure 13. Oct 10 newsletter discussion items



Discussion items

Time	Item	Who	Notes
	External Services Connections	@Ravi Bhagavatula	Mailchimp (Newsletter): @Ravi Bhagavatula wants whenever the user signs up for a newsletter he should get a popup with editorial content. (Is considered a Change Request from Neofonie side) @Jim Curran wants this popup to disappear for one month and then re-appear after this period. (Is considered a Change Request from Neofonie side) @Unknown User (nicolae) - checks if it's feasible an when this feature can be provided.
	My Collection	@Unknown User (nicolae)	@Ravi Bhagavatula - we need My Collection to attract the user in order to come back to the website. So he should get a **Just For U** section/collection where he can received suggested products based on what he already collected beforehand. This way he will be drawn to come back and search for more art. (Is considered a Change Request from Neofonie side) @Unknown User (nicolae) - Amazon is doing something similar, so they check what you searched for and will inform you via email as soon as there are variations of those products available. For starters one could analyze what products will be visited most often and present those within newsletters since they will be most probably close to what users search for. This idea is not engine-based yet, though. @Unknown User (nicolae) - will check on effort and feasablitly. @Ravi Bhagavatula - another idea would be to use the **Thank You** page of the Newsletter Sign Up in order to display which products are recommended by artdesigns.com in order to make the user come back more often. (Is considered a Change Request from Neofonie side)

113. As the notes illustrate, additional features suggested by Defendant for the newsletter include the display of the popup with editorial content upon user sign up, functionality to disappear the pop up for one month and then re-appear after this period and implementing functionality that populates the Thank You page of the newsletter sign-up with products recommended by the Website.

114. A skilled software development practitioner clearly understands that each of the suggestions above can result in significant additional architectural, design, development and testing efforts. For instance, in order to disappear a pop-up from being shown to the same customer for 30 days, an authentication component (or alternative methods of recognizing a visitor) needs to be integrated into the newsletter component. Users might be accessing the Website from a multitude of browsers (Google Chrome, Firefox, Safari, Internet Explorer) or a multitude of devices (an iPad, an Android phone, or a laptop) where the temporarily disappearing pop up needs to be tested for any combination of the above. For instance, should the "snooze for a month" functionality work even for a user that is not logged in? If so, an approach such as storing some user information is a browser cookie might be needed to store user's preferences. What if the same user dismisses the pop up on her laptop's Chrome browser and later accesses the Website from her Safari browser on the same laptop? Should cross-browser information sharing be implemented? What if authentication is required? Would be it be ok for the user to keep seeing the pop-up on every new device or browser she uses (since she's not authenticated)? What if there are multiple newsletters on the Website?[45] Would the user be snoozing all newsletters or a specific one from the website?[46]

115. It is important to note that above feature requests are based on a large number of unvalidated hypotheses on behalf of the client. A constant broadening of the scope of a non-critical feature of an MVP product, before it is even used by a single customer, directly contradicts the very nature of using Agile methodologies and building an MVP. As detailed in Section 4.1, the use of Agile methodologies and MVP products are specifically promoted to avoid making the

[45] As suggested in the JIRA ticket https://jira.neofonie.de/browse/ART-692

[46] Some of the above tickets are labeled as "Change Request". However, given numerous disagreements between the two parties as to what should be treated as a bug vs. a change request, it is difficult to know exactly which one(s) of above tickets were expected to be included as part of MVP by Artissimo.

above mistake and prevent significant unnecessary complexity and burden on development resources before getting early user feedback. To illustrate this, one can consider a few alternative scenarios:

- What if very few users end up subscribing to a newsletter?
- What if a large number of users find the frequency of newsletters too high and end up unsubscribing from a newsletter?
- What if recurring display of newsletter pop-up to unregistered users (which will very likely constitute the majority of visitors to the site) annoys them and causes many to not return to the website?
- What if Artissimo finds out there are significantly more effective ways of increasing customer engagement and sales conversion (through better recommendation, better discovery features, curation, and other means)?

116. Staying loyal to true Agile development practices would have significantly reduced the unnecessary allocation of resources to implementing a bloated newsletter feature before capturing user feedback. Agile development methodologies would promote shipping the product with a bare bone (or non-existent) newsletter, *measure* certain performance indicators to *validate* certain hypothesis, before making any significant and additional effort for development of a feature.

117. User feedback might have very well hinted at the need for significant additional investment into better and more complete newsletter functionality or alternatively might have proved other features as much more effective for reaching certain benchmarks (and hence diverting resources to further development of those other features and away from the newsletter). The fact that as of 10/10/2016 there were still numerous ideas flowing, on a complex feature, whose overall

importance or effectiveness is not yet validated is a clear indication of a departure from Agile development best practices.

118. Moreover, Defendant has stated on multiple occasions that "common best practices"[47] have not been met or implemented. In many occasions what Defendant refers to as being implemented on other websites is well beyond any reasonable definition of a "Minimum Viable Product" and are well beyond the agreed scope indicated in the Phase 2 Offer document. Both parties had a clear agreement about why an MVP is being developed as contemplated in the Phase 2 Offer and as agreed by Defendant:

> *"Artissimo would make further development decisions based on testing user behavior on the MVP"*[48]

119. The detailed and growing list of features for the newsletter is not really what would be considered as "contain the minimum feature set for the artdesigns.com business mode in order to ensure the fastest time to market possible."

120. Finally, to the extent that "Defendant denies that it requested any customization other than that the e-commerce website have sufficient information regarding Defendant's product"[49] and that "Neofonie chose this development approach despite Artissimo's requests for out-of-the-box solutions for certain basic items"[50] I disagree. My above analysis illustrates how significant amount of time was dedicated to implement a highly customized behavior for the newsletter even before the launch of the MVP. In my opinion, a non-customized newsletter functionality that was meant for an MVP website would be significantly simpler than what Plaintiff was asked to develop. In contrast to what was discussed above, the scope would be limited to 1) a

[47] Defendant's Answer and Counterclaim, ¶18 and ¶20.
[48] Defendant's Answer and Counterclaim, ¶12.
[49] Defendant's Answer and Counterclaim ¶9.
[50] Defendant's Answer and Counterclaim ¶27.

newsletter subscription component that would be included in a number of pages of the website 2) an unsubscribe link at the bottom of each newsletter sent to the users.

121. I used newsletter subscription as an example to illustrate a far departure from MVP and Agile development as well as a high level of customization for a product feature. However, these issues were not limited to the development of the newsletter and I was able to trace similar issues in other product features such as My Collection.[51]

122. Using the Project Triangle Model Use, it is easy to see how a change in scope, even if mutually agreed by both parties, directly affects the timeline, overall cost and/or the quality of the final product delivered.

123. In summary, my analysis determined that by not conforming to Agile thinking and MVP software development, Defendant caused a negative impact on the launch timeline of the Website.

5.3 Plaintiff Correctly Implemented Industry Best Practices and Processes for Developing the Website

124. In the Defendant's Answer and Counterclaim document, Defendant alleges that "Neofonie failed to implement best practices or otherwise correct its process". I believe there is no merit to this assertion. Based on my detailed analysis of the process implemented by Neofonie and my years of experience implementing complex software projects, it is my opinion that Plaintiff properly chose a widely used software development methodology[52] and correctly implemented its principles through the course of the development of the Website. By rigorously conforming to Agile thinking, Plaintiff had taken steps that significantly and positively contribute to the successful launch of the Website had it been also supported by Defendant.

[51] https://confluence.neofonie.de/display/AR/_03_My+Collection

[52] See discussion in Section 4.1.

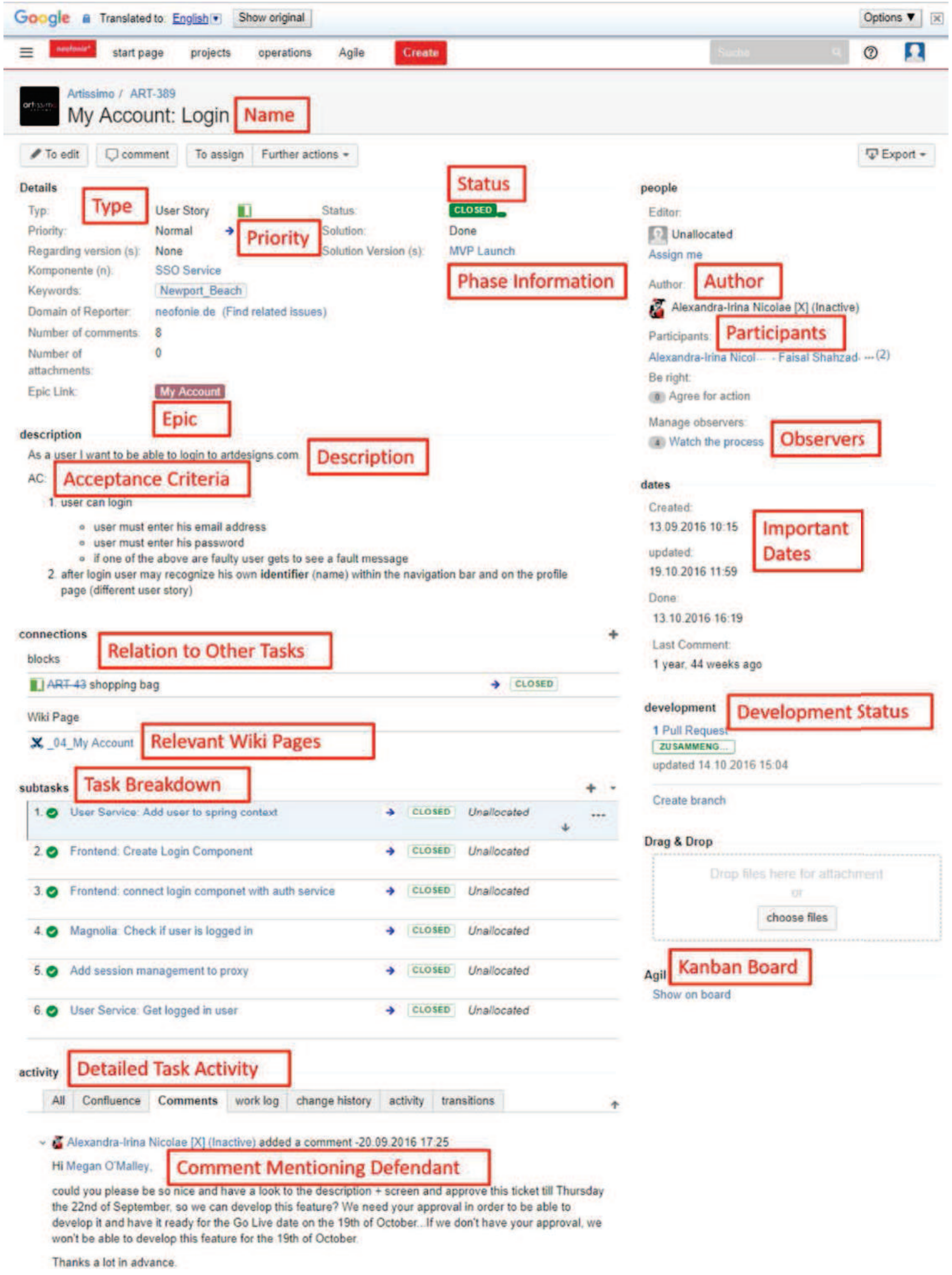
125. As I detailed in Section 4.1 of my report, there are several key components that exist in almost any complex software development project that utilizes Agile methodologies. I studied the process utilized by Plaintiff and observed that these critical components were integral to the process it implemented and closely followed throughout phase 2 of the project:

- Plaintiff had a clear assignment of various roles and responsibilities that included a "Software Architect," "UX Designer," "eCommerce Backend Developer," "Frontend Developer," "Project Leader," "System/DevOps Engineer," and a "QA/Test Engineer." These roles were initially provided to the client[53] and were subsequently assigned various tasks throughout the execution of the project.[54]
- Plaintiff created a defined and documented way of making progress visible in order to minimize subsequent misunderstandings between the promised and developed features for the Website. Plaintiff has accomplished that by creating user stories that map to feature and sub-feature deliverables identified in Section 2.7 of Phase 2 Offer, where each user story includes acceptance criteria defining how one can mark a task as completed. Furthermore, extensive use of JIRA enabled detailed documentation of when each task has started, who has worked on each task, how and when each task was modified, when each task was completed and what epic each task belonged to. Furthermore, upon studying various JIRA tickets, I noticed they were often accompanied by attachments or pointers to wiki pages in Confluence to provide additional information about the nature of each task. The screenshot below illustrates one of the JIRA tickets created for development of the Website.

[53] Phase 2 Offer, Section 2.4 Neofonie Roles.

[54] For instance, see invoice "artissimo AR16_430 IT-Leistungen 08.2016.pdf" and corresponding file "Artissimo_breakdown_August 2016.pdf"

Figure 14. Screenshot from a JIRA ticket created by Plaintiff [55]



Artissimo / ART-389

My Account: Login

Details

Typ: User Story
Priority: Normal
Regarding version (s): None
Komponente (n): SSO Service
Keywords: Newport_Beach
Domain of Reporter: neofonie.de (Find related issues)
Number of comments: 8
Number of attachments: 0
Epic Link: My Account

Status: CLOSED
Solution: Done
Solution Version (s): MVP Launch

description

As a user I want to be able to login to artdesigns.com

AC:

1. user can login
   - user must enter his email address
   - user must enter his password
   - if one of the above are faulty user gets to see a fault message
2. after login user may recognize his own **identifier** (name) within the navigation bar and on the profile page (different user story)

connections

blocks

~~ART-43~~ shopping bag → CLOSED

Wiki Page

_04_My Account

subtasks

1.	User Service: Add user to spring context	→	CLOSED	Unallocated
2.	Frontend: Create Login Component	→	CLOSED	Unallocated
3.	Frontend: connect login componet with auth service	→	CLOSED	Unallocated
4.	Magnolia: Check if user is logged in	→	CLOSED	Unallocated
5.	Add session management to proxy	→	CLOSED	Unallocated
6.	User Service: Get logged in user	→	CLOSED	Unallocated

activity

Alexandra-Irina Nicolae [X] (Inactive) added a comment -20.09.2016 17:25

Hi Megan O'Malley,

could you please be so nice and have a look to the description + screen and approve this ticket till Thursday the 22nd of September, so we can develop this feature? We need your approval in order to be able to develop it and have it ready for the Go Live date on the 19th of October...If we don't have your approval, we won't be able to develop this feature for the 19th of October.

Thanks a lot in advance.

people

Editor:
Unallocated
Assign me

Author:
Alexandra-Irina Nicolae [X] (Inactive)

Participants:
Alexandra-Irina Nicol... · Faisal Shahzad ··· (2)

Be right:
0 Agree for action

Manage observers:
4 Watch the process

dates

Created:
13.09.2016 10:15

updated:
19.10.2016 11:59

Done:
13.10.2016 16:19

Last Comment:
1 year, 44 weeks ago

development

1 Pull Request
ZUSAMMENG...
updated 14.10.2016 15:04

Create branch

Drag & Drop

Drop files here for attachment
or

Agil

Show on board

- Plaintiff performed a detailed analysis to identify project stakeholders in the early stages of the project and has performed more than 10 Stakeholder interviews.[56]

[55] https://jira.neofonie.de/browse/ART-389

- Plaintiff provided a detailed monthly estimation of the resource requirements for each role (Project Leader, Software Architect, UX, etc.) to the client as part of the Phase 2 Offer.[57]
- Plaintiff heavily relied on documentation and creation of extensive wiki pages in Confluence.[58] In addition, it frequently linked various tasks in JIRA to relevant wiki pages in Confluence to provide more context and better understanding of each task.
- Plaintiff participated in numerous meetings with key stakeholders from Artissimo. I studied the Confluence page used by Plaintiff to keep track of meeting notes and identified notes for about 35 meeting taking place from 04/20/16 to 11/14/2016 in the course of about 31 weeks.[59] These meeting notes have detailed information about the stated goals, participants and discussion items as well as additional information when applicable.
- Plaintiff created a detailed QA plan documenting the process of tracking, reporting and monitoring software bugs, software components involved, test strategies, priorities, completion criteria, risk analysis and QA resource requirements.[60] [61]

126. In conclusion, between the recurring meetings, detailed documentation, monitoring features and detailed progress logs of tickets on JIRA, Defendant had been constantly in the loop monitoring progress and receiving updates on how the product was taking shape and was given the opportunity to provide feedback.

[56] https://confluence.neofonie.de/display/AR/Stakeholder+Interviews, https://confluence.neofonie.de/display/AR/KW-Artissimo+Home and https://confluence.neofonie.de/display/AR/Stakeholder+Profiles

[57] Phase 2 Offer, Section 2.6.

[58] https://confluence.neofonie.de/display/AR/KW-Artissimo+Home

[59] https://confluence.neofonie.de/display/AR/Meeting+Notes

[60] https://confluence.neofonie.de/display/AR/How+to+Set+Up+an+Issue+in+JIRA

[61] https://confluence.neofonie.de/display/AR/QA+Plan

5.4 Plaintiff Appropriately Used and Integrated Existing Components into the Website

127. I studied page 13 of the Defendant's Answer and Counterclaim document where Defendant claims Plaintiff accrued unnecessary hours in developing the Website. In particular, Defendant states that "Neofonie decided to develop many basic aspects of the artdesigns.com site from scratch rather than implementing standard non-custom industry standards. These items include but are not limited to the thumbnail carousel, user account management, tax calculation, address validation, and responsive layout."[62] Defendant also states that "Neofonie had access to prior development work that made these development hours unnecessary."[63] In the same section, Defendant also states that "Moreover, Neofonie chose this development approach despite Artissimo's request for out-of-the-box solutions for certain basic items". My analysis showed that Plaintiff did appropriately use and integrate existing components into the Website. Furthermore, Plaintiff had good reasons for not using third-party components in some circumstances. I now offer more details to support my opinion.

128. First, I do not understand what "implementing standard non-custom industry standards" exactly means. Implementing custom standards is orthogonal to developing features from scratch or using OOTB solutions. While it is common for certain product features to follow widely practiced industry standards, on many occasions those standards are met by developing custom code. In fact, it is at times impossible to rely on a disparate set of OOTB software components, developed by different organizations for different purposes, and conform to state of the art or widely accepted industry standards. To the contrary, meeting certain standards would require developing custom code. Second, I do not believe that Plaintiff's "access to prior development work" would make Plaintiff's "development hours unnecessary". Third, I believe

[62] Defendant's Answer and Counterclaim, ¶25.

[63] Defendant's Answer and Counterclaim, ¶26.

Plaintiff has employed the right balance between developing custom software vs. utilizing/integrating COTS and OOTB software components. I now proceed to elaborate on these opinions.

129. I studied tickets created in JIRA as well as relevant documentation in Confluence to understand if Plaintiff had utilized any commercially off-the-shelf product and services (COTS) or out-of-the-box components (OOTB) in the course of design and development of the Website. Based on my findings, I believe that Plaintiff has used integrated or included numerous commercially available software components to the website were applicable. In particular, Plaintiff has identified at several external software components and services to be integrated into the Website rather than developing them from scratch.[64] These components include services related to newsletters (Mailchimp[65]), payment (PayPal[66] and Payeezy[67]), user tracking (Google Analytics[68]), shipment and parcel tracking (Fedex[69]), sending emails (Mandrill[70]) and calculating taxes (Taxjar[71]).

130. As discussed in Section 4.4.2, excessive and blind integration of OOTB software components in a complex software product can significantly increase the overall cost of the product development as well as the overall risk associated with completing the project in time. In the context of developing the Website for Defendant, excessive use of OOTB software could hinder project success and cause significant cost and risk increase because:

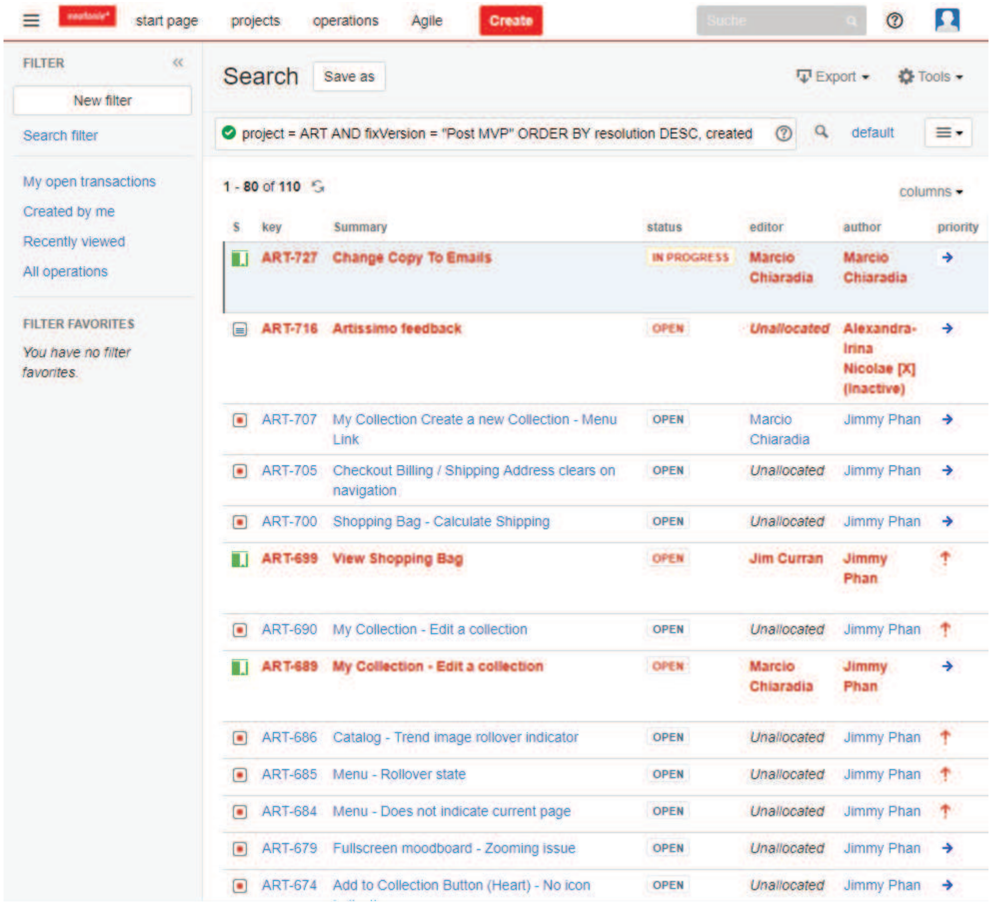
[64] https://confluence.neofonie.de/display/AR/5.+Interfaces
[65] https://confluence.neofonie.de/display/AR/Newsletter
[66] https://confluence.neofonie.de/display/AR/PayPal
[67] https://confluence.neofonie.de/display/AR/Credit+Card
[68] https://confluence.neofonie.de/display/AR/Google+Analytics
[69] https://confluence.neofonie.de/pages/viewpage.action?pageId=20353426
[70] https://confluence.neofonie.de/display/AR/Email
[71] https://confluence.neofonie.de/display/AR/Taxes

- It would increase the number and severity of incompatibility and interoperability issues between various product components. This directly would turn into greater risks for the project as well as additional cost to stitch these packages together.
- It would make it significantly harder (and at times impossible) to meet Defendant requirements due to inability of making any significant changes in the behavior of these software components.
- It would make it significantly more difficult, risky and costly to maintain developed code due to having little control about the level of support provided or effort taken to fix security vulnerabilities of OOTB software components. As I discussed in Section 4.1, maintenance is the single most expensive phase of a vast majority of software projects. Therefore, any decrease in development time and resources needed that comes at the cost of an increase in maintaining the developed product, would almost always have a significant net negative effect in overall project cost and success.

131. As discussed in Section 4.1, one of the key principles of using Agile methodologies to develop an MVP is to delay significant future enhancements, revisions and changes in the product until user feedback is gathered and certain hypotheses are validated. This also means that an MVP product will, by definition, go through significant further development and revision.

132. As the following screenshot from JIRA illustrates, the large number of tickets labeled as “Post MVP” is a clear indication of such intention. Therefore, when dealing with an MVP product, it is even more important to invest in architecting and developing a product that is maintainable and can be significantly enhanced so it can grow into a more complete product over time. Not taking this important point in mind can easily turn an expensive MVP being developed into a throwaway prototype.

Figure 15. JIRA screenshot showing tickets labeled as "Post MVP"



project = ART AND fixVersion = "Post MVP" ORDER BY resolution DESC, created

1 - 80 of 110

S	key	Summary	status	editor	author	priority
▮	ART-727	Change Copy To Emails	IN PROGRESS	Marcio Chiaradia	Marcio Chiaradia	→
☰	ART-716	Artissimo feedback	OPEN	*Unallocated*	Alexandra-Irina Nicolae [X] (Inactive)	→
▣	ART-707	My Collection Create a new Collection - Menu Link	OPEN	Marcio Chiaradia	Jimmy Phan	→
▣	ART-705	Checkout Billing / Shipping Address clears on navigation	OPEN	*Unallocated*	Jimmy Phan	→
▣	ART-700	Shopping Bag - Calculate Shipping	OPEN	*Unallocated*	Jimmy Phan	→
▮	ART-699	View Shopping Bag	OPEN	Jim Curran	Jimmy Phan	↑
▣	ART-690	My Collection - Edit a collection	OPEN	*Unallocated*	Jimmy Phan	↑
▮	ART-689	My Collection - Edit a collection	OPEN	Marcio Chiaradia	Jimmy Phan	→
▣	ART-686	Catalog - Trend image rollover indicator	OPEN	*Unallocated*	Jimmy Phan	↑
▣	ART-685	Menu - Rollover state	OPEN	*Unallocated*	Jimmy Phan	↑
▣	ART-684	Menu - Does not indicate current page	OPEN	*Unallocated*	Jimmy Phan	↑
▣	ART-679	Fullscreen moodboard - Zooming issue	OPEN	*Unallocated*	Jimmy Phan	→
▣	ART-674	Add to Collection Button (Heart) - No icon	OPEN	*Unallocated*	Jimmy Phan	→

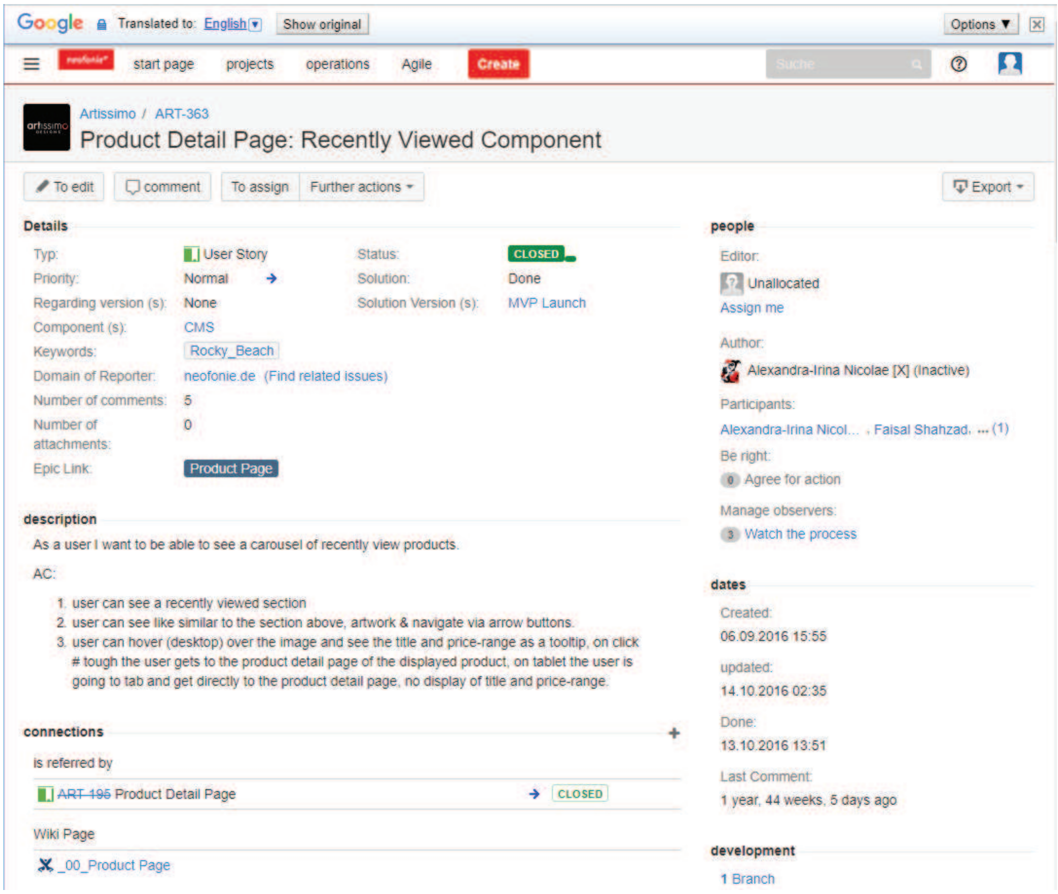
133. Now let's assume Plaintiff had previously implemented some of the features it spent time to develop for the Website. There are still several reasons why Plaintiff could not simply "build from the work it had already done."[72]

134. First, even if similar features were developed by Plaintiff for former clients, such features were likely developed for different products, using different technologies, following a different software architecture, demonstrating similar but not identical behavior, and addressing different requirements and specifications. For instance, I studied the approach taken by Plaintiff

[72] Defendant's Answer and Counterclaim, ¶26.

for the development of the "carousel feature". In my research in JIRA, I identified what I believe is the task that addresses what Defendant is referring to as a "thumbnail carousel" that it alleges Plaintiff developed from scratch.[73] It states:

Figure 16. JIRA task related to the thumbnail carousel[74]



Artissimo / ART-363

Product Detail Page: Recently Viewed Component

Details

Typ: User Story
Priority: Normal
Regarding version (s): None
Component (s): CMS
Keywords: Rocky_Beach
Domain of Reporter: neofonie.de (Find related issues)
Number of comments: 5
Number of attachments: 0
Epic Link: Product Page

Status: CLOSED
Solution: Done
Solution Version (s): MVP Launch

description

As a user I want to be able to see a carousel of recently view products.

AC:

1. user can see a recently viewed section
2. user can see like similar to the section above, artwork & navigate via arrow buttons.
3. user can hover (desktop) over the image and see the title and price-range as a tooltip, on click # tough the user gets to the product detail page of the displayed product, on tablet the user is going to tab and get directly to the product detail page, no display of title and price-range.

connections

is referred by

~~ART-195~~ Product Detail Page CLOSED

Wiki Page

_00_Product Page

people

Editor:
Unallocated
Assign me

Author:
Alexandra-Irina Nicolae [X] (Inactive)

Participants:
Alexandra-Irina Nicol... , Faisal Shahzad, ... (1)

Be right:
0 Agree for action

Manage observers:
3 Watch the process

dates

Created:
06.09.2016 15:55

updated:
14.10.2016 02:35

Done:
13.10.2016 13:51

Last Comment:
1 year, 44 weeks, 5 days ago

development

1 Branch

135. As the above figure illustrates, the task entailed implementing two different behaviors depending on the form factor of the end-user device. On desktop, it required enabling a user to view and *navigate* artwork *via arrow buttons*, *hover over the image* and *see the title and price-range* as a *tooltip* and upon *clicking* on an image get to *the product detail page* of the product

[73] https://confluence.neofonie.de/display/AR/_00_Product+Page
[74] https://jira.neofonie.de/browse/ART-363

displayed. On a tablet, no title or price range would be shown and tapping on an item would take users directly to the product detail page.[75]

136. There are several important compatibility questions that arise in the process of re-using an already developed carousel feature for the Website, assuming it did in fact exist:

- Was the existing carousel feature developed to contain artwork items identical or nearly similar to the website? Although carousels look similar at first glance, a carousel used to show high resolution artwork has a remarkably different set of requirements compared to one created for (say) displaying customer logos. Differences in average file size for each item, the resolution of each image, the total number of images to display, the total number of images visible at any given time and any animation/transition effect in showing carousels would render an already developed carousel component completely useless for another use case.
- Was the existing carousel feature developed in the same front-end programming language as the one used by Plaintiff for the website? An identical carousel, developed in a different programming language or using a different technology would render completely useless for the Website due to language incompatibility issues.
- Was the existing carousel feature developed using the same development framework? For instance, if Website software components were developed using the popular Model-View-Controller framework[76], significant modifications to a non-MVC carousel module would be required before integrating it to the rest of the framework.

[75] https://jira.neofonie.de/browse/ART-363
[76] MVC is one of the most commonly patterns used for development of web-based and other types of software. MVC promotes a logical separation of key components of an application into three groups where model encapsulates information about application data, view captures data being presented to the user and controller connects user input to the other two components.

- Was the existing carousel feature capable of handling the device-specific user requirement for the Website (i.e., supporting arrow keys navigation and rendering specific data fields on hover over effect on desktop, and alternative ways of navigating the thumbnails such as using a swipe gesture on a tablet)? Sometimes, modifying a "close enough" software component to behave in a specific manner takes as much (or at times even more) effort than developing the same component from the ground up. In fact, in certain cases it might be completely infeasible to make such modifications regardless of the amount of time spent.
- Was the existing carousel feature capable of conforming to other important functional requirements specified by the client? For instance, Defendant had indicated the need for "responsive layout" [77] and "localization"[78] for the Website. Again, an already developed carousel that looks and behaves as the client expected on desktop for English language might become completely useless if it required supporting responsive layout or multiple languages on a tablet device.

137. As someone in charge of managing software teams who has extensive background in leading the development of complex software development projects, I have firsthand experience with complexities and nuances of integrating OOTB or COTS packages into existing software products. Very recently, we needed to integrate a responsive logo carousel into our company website. We originally used a widely popular and well-reviewed client logo carousel with over 10,000 active installations.[79] However, after spending significant amount of time and money and even working with the developer of the popular "plugin", both parties determined that there are

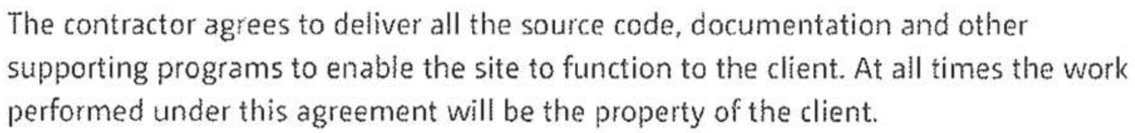
[77] See https://jira.neofonie.de/browse/ART-459 and Defendant's Answer and Counterclaim ¶25 as examples.
[78] See Preamble and Section 2.7 of Phase 2 Offer and https://jira.neofonie.de/browse/ART-308
[79] https://wordpress.org/plugins/gs-logo-slider/

several incompatibility issues between the plugin and the rest of the website. The developer was ultimately unable to properly fix the issue, I ended up hiring a front-end developer that implemented a custom solution from scratch that works properly with the rest of the website.

138. While the above discussion illustrates the compatibility and interoperability issues of integrating a previously developed thumbnail carousel software component into the Website, similar issues could arise and needed to be addressed when other existing software modules for user account management, tax calculation, address validation, and responsive layout were to be integrated into the Website.

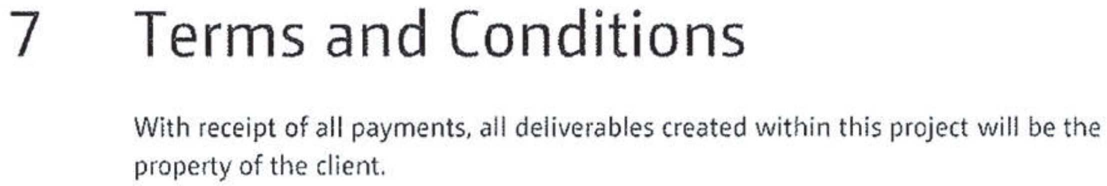
139. There is yet another important consideration for the Plaintiff for using "work it had done previously". It is common practice for established and well-respected software development firms to give full ownership of the intellectual property developed to the client for the product developed. I studied Phase 2 Offer document and noticed it does contain a similar clause enabling Defendant to retain the IP of the work developed effectively preventing Plaintiff from reusing any of the IP developed for this project for future development work.

Figure 17. IP ownership of work created by Plaintiff according to Phase 2 Offer



The contractor agrees to deliver all the source code, documentation and other supporting programs to enable the site to function to the client. At all times the work performed under this agreement will be the property of the client.

Figure 18. Another IP ownership clause reflected in Phase 2 Offer



7 Terms and Conditions

With receipt of all payments, all deliverables created within this project will be the property of the client.

140. It is reasonable to assume Plaintiff had entered into other contracts with similar or even more restrictive IP ownership clauses which would have prevented it from reusing the IP developed for earlier clients for the development of the Website.

141. In summary, out-of-the-box software packages provide useful functionality that can be reused by developers in a variety of applications, reducing the quantity of the code that must be developed from scratch. However, for complex software projects, none of these open source components, either individually or in combination, provides all the functionality of custom-written code. The sheer volume, diversity and specificity of hundreds of JIRA tickets created for the development of the Website indicate the need for a non-generic and fairly complex website to be built. Such project characteristics would have made it even more unlikely for Plaintiff to stitch together OOTB components and achieve client's stated goals for the Website. In my opinion, Plaintiff has consciously decided to integrate the right amount of OOTB and COTS packages to strike a balance between overall development time, final product quality, and maintainability.

142. In addition, contrary to Defendant's belief, use of out-of-the-box solutions is not always possible, feasible or permissible and their excessive use might very well go against the stated goals for project success even for simple software projects.

5.5 Defendant's Portrayal of Severity of Bugs and Plaintiff's Treatment of Them Is Not Accurate

143. Defendant on multiple occasions has cited concerns over the quantity and severity of bugs as well as Plaintiff's lack of willingness to address them as the basis for ending the parties' relationship. I however disagree with this portrayal for at least the following reasons.

144. In the world of complex software development, there is no such thing as a bug-free software. Regardless of the number of hours the development team spends on the QA process, there are two key phases where many more software bugs are discovered: 1) when the customer starts testing the product 2) when the end-user starts using the product. Since all the bugs Defendants has been concerned with are identified in phase 1 (i.e., the Website was never launched), I turn my attention to the former case (bugs found by the Defendant).

145. Once the client begins testing the product, it is common that it discovers more bugs that were not identified by the development team. This happens for a number of reasons.

146. First, regardless of the amount of documentation, the client is ultimately the entity with the most in-depth knowledge of its needs, business processes, user expectations, brand perception, and more. The development team uses a multitude of tools such as storyboards, stakeholder interviews, requirement specifications, documentation and more to best capture these requirements. However, there are often gaps between customers' vision of a product and what gets implemented.

147. Second, the amount of software bugs is correlated with the number and complexity of the software components being developed. As the number and complexity of these components increase, there are more edge cases to be tested, more performance issues that might arise, and it is more likely for two components not to interact with each other the exact way the client expects

them to behave. Therefore, it is more effective and less cost prohibitive for the client to play an active role in the software quality assessment process. I believe adopting Agile methodologies for this project was a critical choice that if adhered by Defendant, would have lent itself well to early identification of many such bugs.

148. Third, and more specific to this case, upon the request of Defendant, the number of hours spent on QA by Plaintiff was significantly cut down.[80] Therefore, it is not surprising for the Defendant to have found several bugs in what Plaintiff had delivered.

149. Lastly, and perhaps most importantly, it is my opinion that certain Defendant actions (or lack thereof) have caused significant delays in completing the development of the Website. Given the importance of this item, I have addressed it separately in Section 5.6 below.

150. Moreover, the Defendant states that "Neofonie's failure to perform under the Agreement resulted in a delay of approximately one year in implementing Artissimo's e-commerce site".[81] I studied the Phase 2 Offer document that suggests the entire engagement between the two parties occurred from early Feb to Nov of 2016. The target launch date in the contract signed by both parties on 05/23/2016 indicates Sep 19th 2016 as the launch date for the MVP. By Oct 21, 2016, the project was almost nearing completion (even by the Defendant's account although it incorrectly believed the project was less close to completion than the Plaintiff.)[82]

151. A delay of about a month is very common for a project of this magnitude, even under perfect synergy between the stakeholders and no chance in scope, which clearly did not exist in this project. Given the inherent complexity of such software projects, they very often risk facing

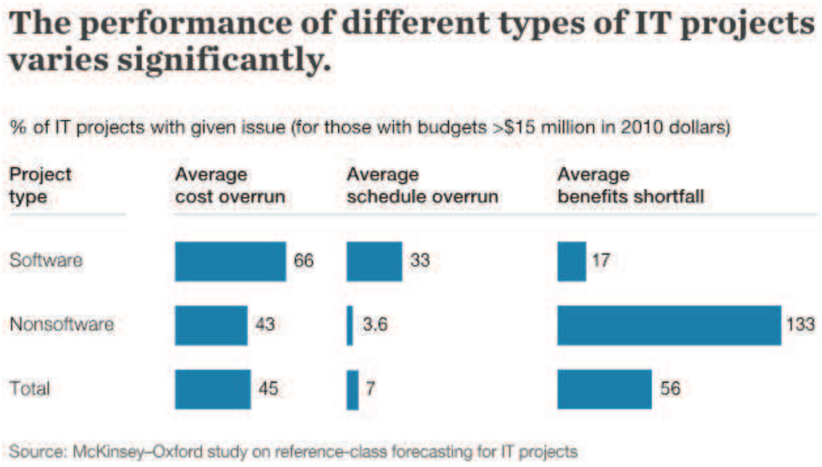
[80] See ART – 007688 and ART – 064695.

[81] Defendant's Answer and Counterclaim, ¶43.

[82] Projection completion is not determined by blindly dividing the number of open tickets by the total number of tickets. Completion is typically measured based on criticality and level-of-effort (LOE) for open items by measuring them in terms of man-hours. See the exchange between Defendant and Plaintiff reflected in ART - 007353 for more details.

massive overruns in budget, time or both. In a detailed study, McKinsey & Company and the University of Oxford analyzed the budgets, schedules and predicted performance benefits of more than 5,400 IT projects. Among many other findings, the study determined that "half of all large IT projects—defined as those with initial price tags exceeding $15 million—massively blow their budgets. On average, large IT projects run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted. Software projects run the highest risk of cost and schedule overruns."[83]

Figure 19. Average cost and schedule overrun for software projects[84]


The performance of different types of IT projects varies significantly.

% of IT projects with given issue (for those with budgets >$15 million in 2010 dollars)

Source: McKinsey–Oxford study on reference-class forecasting for IT projects

152. According to various ticket updates in JIRA and communication between the two parties, Plaintiff continued to actively resolve even more issues between Oct 21 and Nov 23, where Defendant announced it was ending the relationship with Plaintiff. Therefore, a person skilled in the world of software development would agree that regardless of the root cause, the delay in

[83] Delivering large-scale IT projects on time, on budget, and on value. Michael Bloch et al., McKinsey & Company https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value

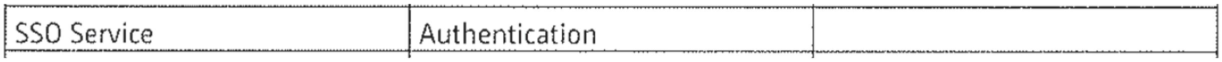
[84] Delivering large-scale IT projects on time, on budget, and on value. Michael Bloch et al., McKinsey & Company https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value

reaching project completion was relatively small and that Defendant did not suffer a "delay of approximately one year in implementing Artissimo's e-commerce site."

153. As yet another example of incorrect portrayal of facts about the nature and existence of bugs I studied the issue of authentication. As part of Defendant's allegations regarding "Neofonie's Inadequate Performance of Phase Two", Defendant states that "Throughout the Project, several basic ecommerce deliverables were not implemented properly, including but not limited to tax calculation, address validation, user logout, edit checkout order details, dynamic and responsive layout, and the add-to-cart notification."[85] Defendant refers to the user logout feature yet again in another document stating that "The website that Neofonie created in Phase Two was not viable because […] the filtering and logoff functions were not operative".[86]

154. I studied the Phase 2 Offer to identify the user requirements for the logout feature. Section 2.7 of that document provides the following scope for the logout feature:

Figure 20. Scope of authentication according to deliverables for Phase 2 Offer



SSO Service	Authentication	

155. I employed the software process assessment approach discussed in Section 4.5.2 and studied JIRA and other relevant case documents to identify the status of the logout feature and to learn whether Plaintiff resisted implementing this feature or treated it as a change request (i.e., beyond the scope of MVP). Thanks to a well-established process by Plaintiff and detailed information available on JIRA, I was able to find information contradicting these claims.

156. According to JIRA logs, Plaintiff was planning to implement the logout feature as part of the MVP launch. In a comment left on 10/13/2016 for the JIRA ticket titled "My Account:

[85] Defendant's Answer and Counterclaim, ¶21.

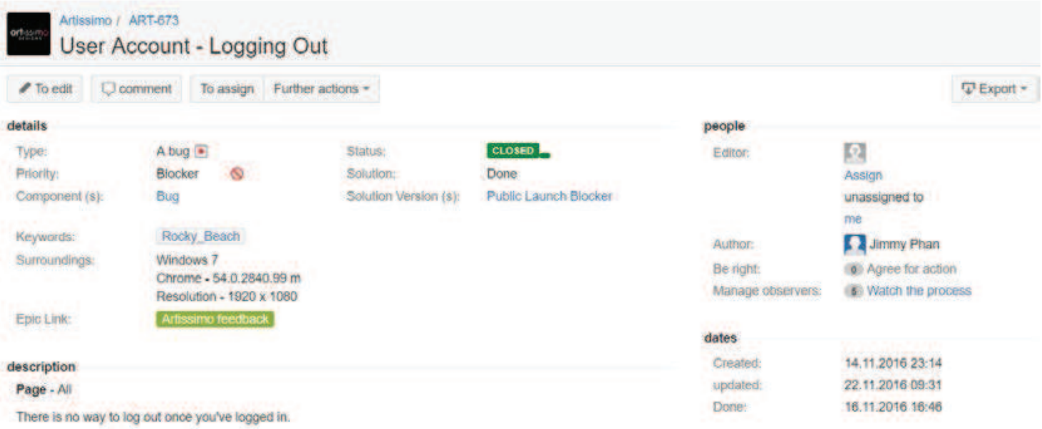
[86] Artissimo Designs LLC's Responses to Neofonie GMBH's First Set of Interrogatories, page 3.

Login”, it is stated that “A logout button is not planned / implemented yet. If you want to do a logout restart your browser or delete the auth cookie with the browsers deveveloper tools.”[87]

157. Similarly, in the excel spreadsheet titled ArtDesign-Issues-2016_11_14.xlsx, Neofonie admits that this is a bug. The note under “Notes Neofonie” states “True – Bug” for the corresponding row. Moreover, this ticket is not listed under change request in JIRA.

158. I found another JIRA ticket created on 11/14/2016 titled “User Account – Logging Out” specifically to address this bug. The ticket type is “A Bug” and not a “Change Request” and the priority is set to “Blocker”.[88] The following screenshot illustrates more details about this ticket.

Figure 21. Screenshot from the logout JIRA ticket



Artissimo / ART-673

User Account - Logging Out

details

Type: A bug
Priority: Blocker
Component (s): Bug
Status: CLOSED
Solution: Done
Solution Version (s): Public Launch Blocker
Keywords: Rocky_Beach
Surroundings: Windows 7
Chrome - 54.0.2840.99 m
Resolution - 1920 x 1080
Epic Link: Artissimo feedback

description

Page - All

There is no way to log out once you've logged in.

people

Editor: Assign unassigned to me
Author: Jimmy Phan
Be right: 0 Agree for action
Manage observers: 5 Watch the process

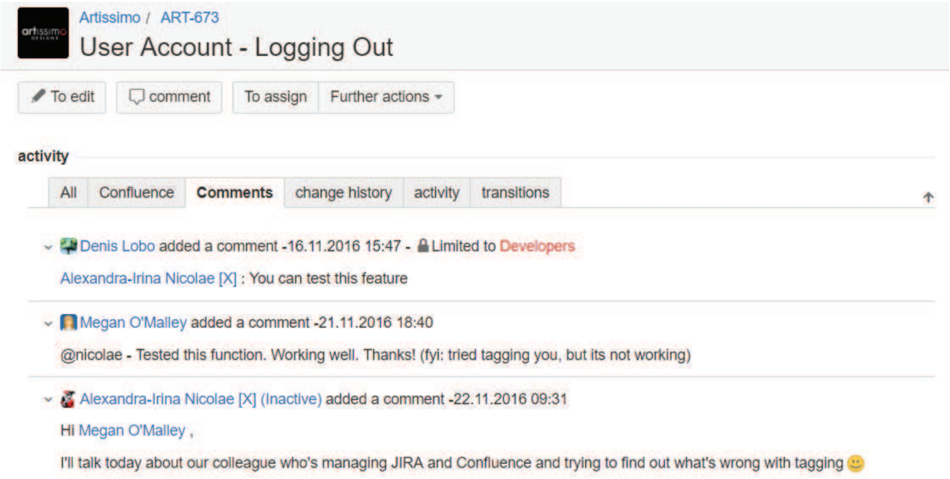
dates

Created: 14.11.2016 23:14
updated: 22.11.2016 09:31
Done: 16.11.2016 16:46

159. I studied the “Comments” section of the above ticket. I found out a comment is left by Plaintiff on 11/16/2016 stating that this feature is ready to be tested. This comment is followed by another comment made by Megan O’Malley, one of Defendant employees, a few days later on 11/21/2016 admitting that she has in fact “Tested this function” and that it is “Working well”. The following screenshot from the same ticket discussed above illustrates this exchange.

[87] https://jira.neofonie.de/browse/ART-389
[88] https://jira.neofonie.de/browse/ART-673

Figure 22. Comment section of the logout JIRA ticket



Artissimo / ART-673

User Account - Logging Out

activity

Denis Lobo added a comment -16.11.2016 15:47 - Limited to Developers

Alexandra-Irina Nicolae [X] : You can test this feature

Megan O'Malley added a comment -21.11.2016 18:40

@nicolae - Tested this function. Working well. Thanks! (fyi: tried tagging you, but its not working)

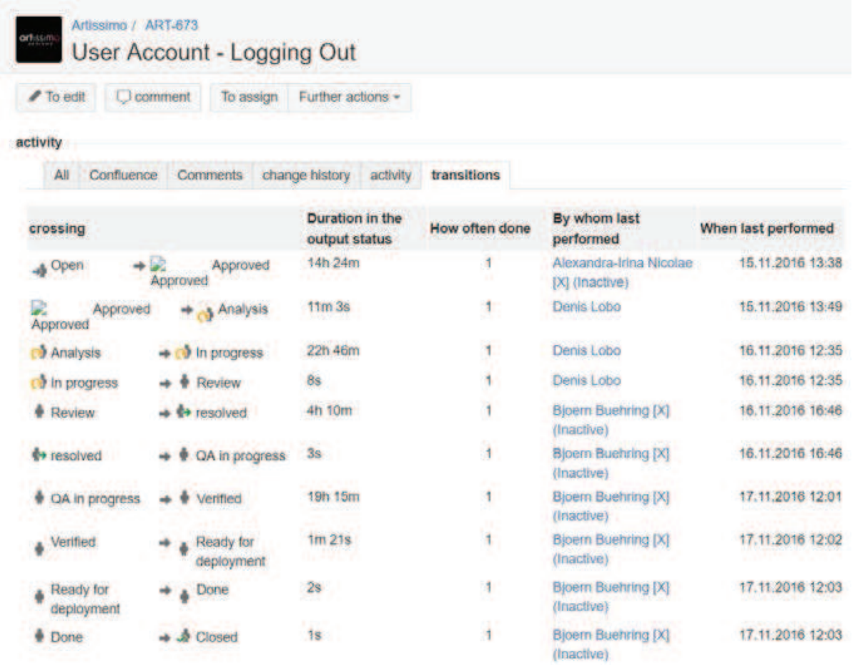
Alexandra-Irina Nicolae [X] (Inactive) added a comment -22.11.2016 09:31

Hi Megan O'Malley ,

I'll talk today about our colleague who's managing JIRA and Confluence and trying to find out what's wrong with tagging 🙂

160. Finally, the following screenshot illustrates the "Transactions" tab of the same JIRA ticket that provides more details about the status of the ticket between November 15 to November 17 of 2016.

Figure 23. Transaction section of the logout JIRA ticket



Artissimo / ART-673

User Account - Logging Out

activity

crossing		Duration in the output status	How often done	By whom last performed	When last performed
Open	→ Approved Approved	14h 24m	1	Alexandra-Irina Nicolae [X] (Inactive)	15.11.2016 13:38
Approved Approved	→ Analysis	11m 3s	1	Denis Lobo	15.11.2016 13:49
Analysis	→ In progress	22h 46m	1	Denis Lobo	16.11.2016 12:35
In progress	→ Review	8s	1	Denis Lobo	16.11.2016 12:35
Review	→ resolved	4h 10m	1	Bjoern Buehring [X] (Inactive)	16.11.2016 16:46
resolved	→ QA in progress	3s	1	Bjoern Buehring [X] (Inactive)	16.11.2016 16:46
QA in progress	→ Verified	19h 15m	1	Bjoern Buehring [X] (Inactive)	17.11.2016 12:01
Verified	→ Ready for deployment	1m 21s	1	Bjoern Buehring [X] (Inactive)	17.11.2016 12:02
Ready for deployment	→ Done	2s	1	Bjoern Buehring [X] (Inactive)	17.11.2016 12:03
Done	→ Closed	1s	1	Bjoern Buehring [X] (Inactive)	17.11.2016 12:03

161. In summary, the aforementioned comments and activities by Plaintiff suggest that Defendant's claim that Plaintiff repeatedly referred to items such as the logout functionality as "beyond the scope of the Agreement" as well as Defendant's claim that they "were not implemented properly" or "were not operative" are incorrect. They also show that Plaintiff did not refuse to address bugs related to MVP features of the Website.

5.6 Defendant's Actions (or Lack Thereof) Significantly Contributed to Delays in Launching the Website

162. My analysis of the available evidence such as several JIRA tickets and communication between Defendant and Plaintiff suggest that the Defendant caused unnecessary delays in launch of the Website.

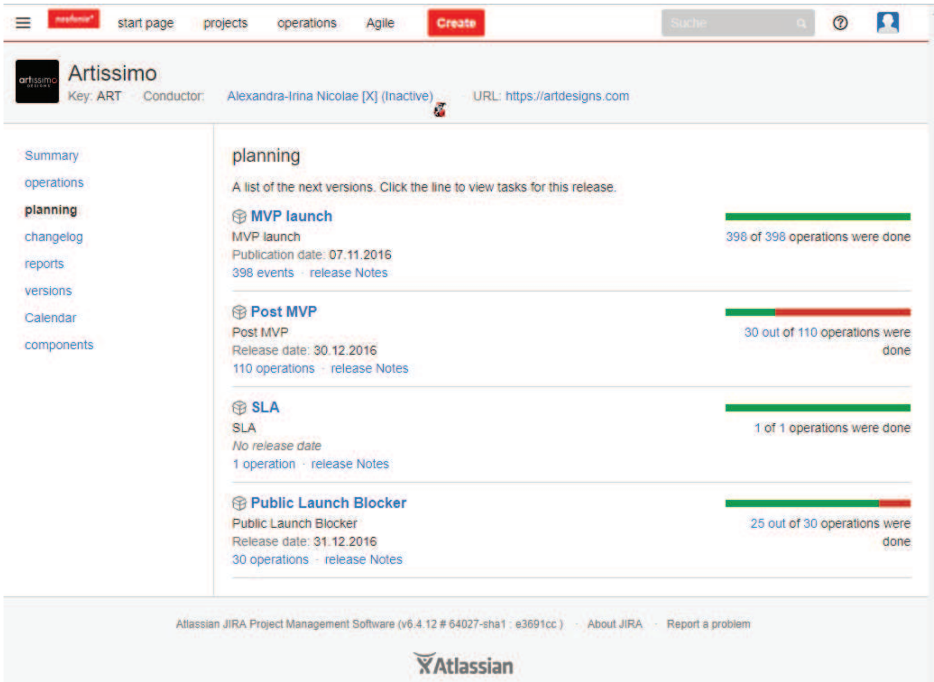
163. In Section 4.1 of my report, I presented an in-depth discussion on the importance of early testing. I discussed how overall software project cost and risk of failure can greatly increase by deferring and delaying QA to later stages of the development lifecycle. I also discussed how the advent of modern software development methodologies such as Agile reduce overall project risks by creating numerous build-test-evaluate feedback loops where the output of each cycle is critical in accomplishing success in future iterations of product development. I also discussed the importance of closely adhering to the adopted software development methodology used for a project.

164. In the course of designing and developing complex software projects, a well-defined and meticulously practiced methodology for software development directly improves the quality of the final product created using that methodology. In other words, the quality of the final product is directly governed by the quality of the process by which the software was developed. In my opinion, a significant delay in beginning the QA process for the Website, coupled with lack

of conformity to a tried and tested methodology and Defendant's unwillingness or lack of ability to unblock Plaintiff with critical items resulted in a far departure from the above principles.

165. A study of tickets created in JIRA throughout the course of the project lifecycle illustrates a methodical approach by the Plaintiff in maintaining a backlog of tasks to pull from, labeling each task with various stages of the project (MVP Launch, Post MVP, etc.), development and QA of features and constantly engaging Defendant throughout the process. The following screenshot taken from JIRA, provides more details about the breakdown of hundreds of tasks created for the development and launch of the Website.

Figure 24. Task report from JIRA[89]



Artissimo

Key: ART Conductor: Alexandra-Irina Nicolae [X] (Inactive) URL: https://artdesigns.com

planning

A list of the next versions. Click the line to view tasks for this release.

MVP launch
MVP launch
Publication date: 07.11.2016
398 events · release Notes

398 of 398 operations were done

Post MVP
Post MVP
Release date: 30.12.2016
110 operations · release Notes

30 out of 110 operations were done

SLA
SLA
No release date
1 operation · release Notes

1 of 1 operations were done

Public Launch Blocker
Public Launch Blocker
Release date: 31.12.2016
30 operations · release Notes

25 out of 30 operations were done

Atlassian JIRA Project Management Software (v6.4.12 # 64027-sha1 : e3691cc) · About JIRA · Report a problem

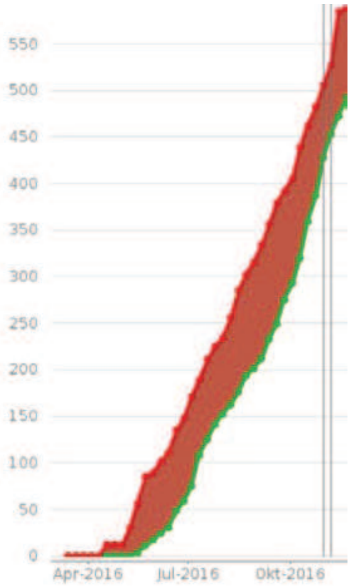
Atlassian

166. The following graph, also generated by JIRA, illustrates the number of tickets created vs. resolved over time.[90] The red and green lines demonstrate the number of tasks created

[89] https://jira.neofonie.de/browse/ART/?selectedTab=com.atlassian.jira.jira-projects-plugin:roadmap-panel
[90] The graph is also referred to as Cumulative Flow Diagram or CFD.

and completed over time, respectively. Therefore, the area between the two lines represents the difference between the number of tasks created vs. the number of tasks completed in JIRA. In other words, it acts as a proxy for how well the development team was able to keep the pace of development with the rate of new tickets being created.

Figure 25. JIRA tasks created (red) vs. completed (green)



167. This graph illustrates a disciplined approach by Plaintiff in actively keeping up with tasks created in JIRA and maintaining a linear flow of development. The graph also illustrates an important characteristic of developing software projects: the slope of the red line is determined by the number of tasks created in a unit of time while the slope of the green line is directly determined by the size of the development team (and how fast they resolve various tasks). Therefore, a significant delay in reporting bugs (hence creating a large number of tickets in a short period of time) causes a huge spike in the red line, which in turn will cause delays in project completion. More importantly, as discussed in Section 4.1, some of these JIRA tasks are blockers to other tasks since they act as a prerequisite and need to be completed before development of other tasks beings.

Similarly, some of these JIRA tasks are sub-tasks of other tasks or have their own prerequisites and some of these tasks take a significantly longer time to resolve. Therefore, a delay in reporting (and hence completing) important tasks materially delays the entire project.[91] As illustrated by The Project Management Triangle[92], in these circumstances the pace of development cannot be arbitrarily increased without incurring more cost, reducing scope and/or degrading the overall quality of the end product.

168. My analysis revealed multiple instances where Plaintiff had little to no control over the delays in completing development of the Website because of Defendant's inability or unwillingness to unblock the Plaintiff in a timely manner.

169. For instance, I studied Section 2.7 of Phase 2 Offer. The agreement states in extremely clear language that "A license for the Content Management System Magnolia and a hosting environment for artdesings.com are must-have prerequisites for the MVP". Based on my analysis of JIRA tickets and relevant email exchanges between the Defendant and Plaintiff, this important prerequisite was not provided to Plaintiff despite numerous requests made to the Defendant in the course of several months.

170. In addition to requesting the Magnolia license on the Phase 2 Offer document, Plaintiff has repeatedly made the same request at least in an email on 07/27/2016 [93], another email on 08/16/2016[94], another email on 09/08/2016[95], a call on 09/20/2016[96], another email on

[91] This concept, also known as the *critical path*, is also very well studied in project management. For instance, see Project Management Institute (2013). A Guide To The Project Management Body Of Knowledge (5th ed.). Project Management Institute. ISBN 978-1-935589-67-9.

[92] See Section 4.2.

[93] NEO 01666

[94] NEO 01687

[95] ART 016954

[96] https://confluence.neofonie.de/display/AR/2016-09-20+Call+with+Megan+and+Jimi

10/18/2016[97], another email on 10/20/2016[98], another email on 10/21/2016[99], another email on 10/25/2016[100], a meeting held on 10/25/2016[101], the meeting held on 11/03[102], another email on 11/07/2016[103] and a meeting on 11/14[104].

171. It is important to note that there were numerous tasks that depend on or were blocked by completion of the above task. A skilled person in the world of software development would agree that migration from a test environment to a live environment entails a large number of activities that typically include but is not limited to making changes to or adding: new certificates, server IP addresses, firewall rules, authentication, DNS records, load-balancing servers, mail servers, cache servers, security configurations, integration with other system components, search engine optimization, integration test, user-acceptance test and much more. It is unclear to me how Defendant blames Plaintiff for delays in project completion where an important prerequisite, identified and mutually agreed as early as the very beginning of phase 2 of the project, was not addressed until the very end of the project.

172. As yet another example I studied addition of products to Microsoft Dynamics AX (AX), a key component being integrated with the Website.[105] The AX component of the Website was meant to store and keep track of various aspects of artdesigns.com business such as customer, payment, product, sales and order data.

173. Here are just some of the instances were Plaintiff had made a request to the Defendant to unblock Plaintiff with necessary information regarding products on AX: a call on

[97] ART 007412
[98] ART 007408
[99] NEO 01804
[100] ART 007364
[101] https://confluence.neofonie.de/display/AR/2016-10-25+Update+call+with+Megan
[102] https://confluence.neofonie.de/display/AR/2016-11-03+Bi+Weekly
[103] ART 006580
[104] https://confluence.neofonie.de/display/AR/2016-11-14+Meeting+Notes+artissimo.com
[105] https://dynamics.microsoft.com/en-us/ax-overview/

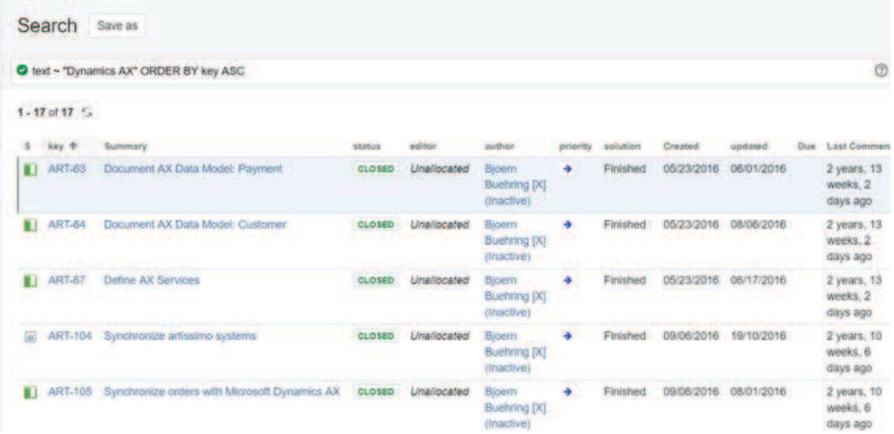
09/29/2016[106], another email on 10/20/2016[107], another call on 10/25/2016[108], an email on 27/10/2016[109], another email on 11/02/2016 where Plaintiff specifically lists "Creation of products and pushing those to AX" as one of the items that "Neofonie needs from Artissimo, but is missing and would cause a further delay of the public launch, if not addressed asap"[110] and the meeting on 11/03/2016.[111]

174. In response to Plaintiff being blocked on this issue for an extended period of time in an email on 11/1/2016, Defendant admits that "WE ARE HAVING ISSUES WITH ADDING PRODUCTS TO TEST ENVIRONMENT. WE EXPECT TO HAVE THIS ADDRESSED IN THE NEXT TWO DAYS".[112]

175. Defendant's lack of cooperation or response to Plaintiff in adding enough products to AX is yet another example of delays being created which were outside of Plaintiff's influence and control. As the screenshot below illustrates, there were at least 17 tickets in JIRA directly related to AX. Similarly, there were numerous documents on Confluence addressing integration issues between AX and the Website. Given the tight coupling of AX and the Website, delays in unblocking Plaintiff with issues related to AX would have a significant impact on completion of tasks depending on Defendant's actions as well as testing numerous other components and features directly affected or related to it.

[106] https://confluence.neofonie.de/display/AR/2016-09-20+Call+with+Megan+and+Jimi
[107] ART 007408
[108] https://confluence.neofonie.de/display/AR/2016-10-25+Update+call+with+Megan
[109] ART 016589
[110] NEO 01825
[111] https://confluence.neofonie.de/display/AR/2016-11-03+Bi+Weekly
[112] ART 006613

Figure 26. Some of JIRA tickets related to Microsoft Dynamics AX



Search Save as

text ~ "Dynamics AX" ORDER BY key ASC

1 - 17 of 17

S	key	Summary	status	editor	author	priority	solution	Created	updated	Due	Last Commen
	ART-63	Document AX Data Model: Payment	CLOSED	Unallocated	Bjoern Buehring [X] (Inactive)	➔	Finished	05/23/2016	06/01/2016		2 years, 13 weeks, 2 days ago
	ART-64	Document AX Data Model: Customer	CLOSED	Unallocated	Bjoern Buehring [X] (Inactive)	➔	Finished	05/23/2016	08/06/2016		2 years, 13 weeks, 2 days ago
	ART-67	Define AX Services	CLOSED	Unallocated	Bjoern Buehring [X] (Inactive)	➔	Finished	05/23/2016	06/17/2016		2 years, 13 weeks, 2 days ago
	ART-104	Synchronize artissimo systems	CLOSED	Unallocated	Bjoern Buehring [X] (Inactive)	➔	Finished	09/06/2016	19/10/2016		2 years, 10 weeks, 6 days ago
	ART-105	Synchronize orders with Microsoft Dynamics AX	CLOSED	Unallocated	Bjoern Buehring [X] (Inactive)	➔	Finished	09/06/2016	08/01/2016		2 years, 10 weeks, 6 days ago

176. A person experienced with the world of software development in general and e-commerce applications in particular would appreciate an extensive set of tasks and components that could not be properly tested if enough products are not made available proper completion of the QA process. For instance, implementing and testing pagination logic require the availability of a large number of products, so that enough pages are created to test navigation between pages as well as ensuring items are displayed in one and only one page. As another example, there are several performance issues such as caching and memory management that are directly affected and can only be properly tested in the presences of a large number of items. As yet another example, filtering, sorting and other search related operations can only be properly tested when enough number of products are available. These are just some of the examples that illustrate how a delay caused by Defendant can easily create a ripple effect in numerous other tasks that need to be completion in a timely manner. Lastly, the blocking issues are not limited to those discussed above and I was able to find several other examples where delays were directly attributed to Defendant's failure to act timely in unblocking Plaintiff on critical issues.

177. To summarize, in contrast to what Defendant alleges, its unresponsiveness, inability or unwillingness to address critical and blocking issues in a timely manner, its decision

to start the QA at later stage of the product lifecycle, its request to reduce the QA efforts suggested by Plaintiff, among other things, would have made the launch date meaningless and gave little control to Plaintiff to ensure timely launch of the Website.

6 CONCLUSIONS

178. In conclusion, my analysis reached the following findings:

- Defendant didn’t conform to Agile and MVP development principles which resulted in unnecessary delays in the development of the Website.
- Plaintiff employed a widely accepted software development process and correctly implemented it in the course of developing the Website.
- Plaintiff’s decision in feasibility and applicability of integrating out-of-the-box components with the website was appropriate and its decision not to excessively integrate such components was justifiable.
- Defendant inaccurately portrayed Plaintiff’s treatment of several bugs as well as their severity.
- Defendant’s failure to take important actions resulted in Plaintiff being unable to perform its duties according to the agreement between the parties for the development of the Website.

I declare under penalty of perjury that the foregoing is true and correct.

Executed on 09/17/2018 at Los Angeles, California.

Ali Khoshgozaran

EXHIBIT A: CURRICULUM VITAE OF ALI KHOSHGOZARAN

See attachment.

ALI KHOSHGOZARAN

Computer Scientist

Quandary Peak Research
205 S Broadway, Suite 300
Los Angeles, CA 90012
Phone: (323) 545-3926
Email: ali@quandarypeak.com

EDUCATION

Ph.D. in Computer Science, University of Southern California	**May 2010**
M.S. in Computer Science, The George Washington University	**May 2005**
B.Sc. in Computer Engineering, Sharif University of Technology	**June 2003**

EMPLOYMENT

- **Computer Scientist - Quandary Peak Research, Los Angeles, CA** **2018**
  - Providing software analysis in software-related litigation patent, copyright infringement, breach of contract and other matters
  - Analyzing software intellectual property and patent portfolios for validity and infringement in the context of licensing and brokering negotiations, startup investments, and M&A
  - Investigating software failures to determine the root cause and help clients understand whether and how the failure could have been avoided
  - Documenting the architecture of software systems to identify structural similarities and differences among competing products and services and deduce the origin of software designs and code

- **Founder & CEO - Tilofy Inc, Los Angeles, CA** **2013-2017**
  - Five years of hands-on technical execution, leadership, product architecture and strategy
  - Designed, architected and led the development of Tilofy's complex technology infrastructure and software stack
  - Successfully built and released one of the most disruptive products in the predictive analytics to the market from the ground up
  - Interviewed 80+ candidates and managed a team of 15 backend/frontend/iOS engineers, biz devs, product managers & analysts
  - Led data/service integration of Tilofy with Twitter (Gnip), Google (GCE), IBM (Watson), Microsoft (Azure) & Amazon (AWS)
  - Performed daily code reviews, design discussions, bug tracking and product development with the engineering team
  - Turned a $20K seed investment into a cash-flow positive technology business worth $9M (raised $2M+ of venture funding)
  - Personally signed up world's biggest brands as paid clients (Red Bull, Altria, Hershey, Henkel, Unilever, Samsung, Anheuser Busch, etc.)

- Interfaced directly with heads of innovation, analytics, marketing, consumer insights and the CXO offices of Tilofy clients
- Technologies used: ElasticSearch/MySql (storage), Ruby/Java (backend/APIs), React/Angular (frontend), AWS/Rackspace/GCE (IaaS), Jenkins/Chef/Salt/Kibana/Github/Log4j (tools), Mixpanel/Cloudflare/Google Analytics (analytics), Scrum (methodology)

- **Technical Lead, Samsung SmartTV Platform, Irvine, CA 2010-2013**
  - Led technical integration of partners micro-services (Turner, TWC, Verizon) with Syncplus for Samsung's North American release
  - Led the innovation team in charge of conceptualizing and planning new product ideas for Samsung SmartTVs 2-5 years out
  - Developed proof of concept prototypes for Samsung's senior management to identify product commercialization opportunities
  - Filed 10 patents with Samsung on proprietary technology and related use cases (3 granted)

- **Program Manager Intern, Microsoft Online Services Division, Redmond, WA 2009**
  - Delivered and owned specs for Bing Maps "event app"
  - Worked directly with external/internal software engineers on design and development of an event app for Bing Maps platform
  - Responsible for ingestion, indexing and integration of Microsoft partner's events data into Bing Maps data layer
  - Selected among very few employees to be mentored 1:1 by Microsoft SVPs as potential future leaders for the organization
  - Received but declined a full-time offer

- **Software Developer Intern, Yahoo! Behavioral Targeting , Sunnyvale CA 2007**
  - Designed and developed various software products for the behavioral targeting QA team
  - Implemented an end-to-end framework for querying, visualizing and analyzing massive user behavioral data
  - Implemented a web-based interface to simulate user browsing behavior across hundreds of Yahoo! properties
  - Filed a patent with Yahoo! on developed technologies and querying interfaces
  - Received but declined a full-time offer

- **Research Assistant, University of Southern California, Los Angeles, CA 2005-2010**
  - Member of the research team at the Information Laboratory group within the Computer Science department
  - Software designer/developer for the GEODEC (geospatial decision-making) project funded by grants from Microsoft, Google, NSF, etc. The goal of the project was to construct an information-rich and realistic 3-dimensional visualization and simulation of a geographical location, rapidly and accurately
  - Designed and developed a three-layered architecture to import, ingest and analyze large amounts of heterogeneous, multi-modal and real-time spatio-temporal for real-time querying and visualization purposes
  - Developed an activity classification application on Android G1 phones using accelerometer, GPS, camera and other sensor data to predict user activities
  - Designed and developed several location-based services interacting with mobile clients for geospatial decision-making
  - Conducted research on privacy-preserving approaches to providing location based services

- Developed research grant proposals for NSF funding in the areas of geo-spatial information management and location privacy
- Designed and developed the first moving object tracking and querying application using Google Earth featured on MSNBC and Google Earth's blog
- Invented a fundamental space-transformation technique to perform the class of Nearest Neighbor (NN) queries, the core class of queries used in many of the location-based services, without revealing the origin of the query in order to preserve the privacy of this information
- A patent granted on the novel space transformation method and technique proposed

- **Teaching Assistant, University of Southern California, Los Angeles, CA 2005-2006**
  - Mentoring students projects and designing exam questions for the Programming Systems Design course
  - Giving guest lectures and leading TA sessions for Database Information Systems course

- **Research Assistant, The George Washington University, Washington, D.C. 2004-2005**
  - Contributing research towards establishing a framework for Statistical Cryptanalysis, under the supervision of Dr. Poorvi Vora

- **Lab Instructor, The George Washington University, Washington, D.C. 2004-2005**
  - Designing and instructing lab sections of "Database Management Systems", teaching Oracle, MYSQL, JDBC, SQLPLUS, ODBC, XML etc.

- **Teaching Assistant, The George Washington University, Washington, D.C. 2004**
  - Mentoring students in "Group Software Projects" course to build a web-based application for a real client (including social impact, professional ethics and intellectual property analysis of the project).

LITIGATION CONSULTING

- **Certain Infotainment Systems, Components Thereof and Automobiles Containing the Same (Broadcom v. Toyota Motor Corporation, Panasonic Corporation et al.)**
  - Jurisdiction: United States International Trade Commission
  - Counsel: Steptoe & Johnson LLP
  - Nature of Suit: Patent

- **UCAR TECHNOLOGY (USA) INC. and UCAR INC. v. YAN LI et al.**
  - Jurisdiction: Northern District of California
  - Counsel: Ruyak Cherian LLP
  - Nature of Suit: Trade Secret (Machine Learning, Artificial Intelligence, Self-Driving Cars)

- **CAPSTONE LOGISTICS HOLDINGS, INC., et al., v. PEDRO NAVARRETE et al.**
  - Jurisdiction: Southern District of New York
  - Counsel: Ferber Law, A Professional Corporation
  - Nature of Suit: Trade Secret

- **Neofonie GMBH v. Atrissimo Designs LLC**
  - Jurisdiction: Central District of California
  - Counsel: Emanuel Law
  - Nature of Suit: Breach of Contract

- **SEVEN Networks, LLC v. ZTE (USA) Inc. and ZTE Corporation**
  - Jurisdiction: Northern District of Texas
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent Infringement

- **Hitachi Maxell, LTD. v. ZTE Corp. and ZTE USA Inc.**
  - Jurisdiction: Eastern District of Texas
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent Infringement

- **Alec Farwell v. Edward J. Herzstock and Revleap Corp.**
  - Jurisdiction: Superior Court for California, County of Los Angeles
  - Counsel: Wolk & Levine LLP
  - Nature of Suit: Corporate Civil Suit

- **CYWEE Group LTD. v. ZTE Corporation, ZTE (USA), Inc. and ZTE (TX) Inc.**
  - Jurisdiction: Southern District of California
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent Infringement

- **Cellular Communications Equipment LLC v. HTC America Inc., HTC Corporation, ZTE Corporation, ZTE Solutions, Inc. and ZTE USA, Inc.**
  - Jurisdiction**:** Eastern District of Texas
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent Infringement

- **Bryndon Fisher v. The United States of America**
  - Jurisdiction: United States Court of Federal Claims
  - Counsel: Schubert Jonckheer & Kolbe LLP
  - Nature of Suit: Class Action

PATENTS

- Middle Partners, Dang Van Tran, Yingnan Zhu, Xing Zheng, Jaffar Khoshgozaran. Samsung Electronics Co. LTD. (Granted)

- Apparatus and Methods for Tracking, Querying, and Visualizing Behavior Targeting Processes, Michael J. Menezes, Jaffar Khoshgozaran. Yahoo!, Inc. (Granted)

- Multi-User Discovery, Dang Van Tran, Yingnan Zhu, Xing Zheng, Jaffar Khoshgozaran. Samsung Electronics Co. LTD. (Granted)

- Blind Evaluation Of Nearest Neighbor Queries Wherein Locations Of Users Are Transformed Into A Transformed Space Using A Plurality Of Keys, Cyrus Shahabi, Jaffar Khoshgozaran, Houtan Shirani-Mehr. University of Southern California. (Granted)

- Crowd Sourcing, Dang Van Tran, Yingnan Zhu, Xing Zheng, Jaffar Khoshgozaran. Samsung Electronics Co. LTD. (Filed)

- Privacy and Trends, Jaffar Khoshgozaran, Dang Van Tran, Xing Zheng, Yingnan Zhu. Samsung Electronics Co. LTD. (Filed)

- Augmented Intelligent Context, Yingnan Zhu, Xing Zheng, Daniel P. Gicklhorn, Michael R. Lovelace, Jaffar Khoshgozaran, Dang Tran. Samsung Electronics Co. LTD. (Filed)

- Context-Aware Media Interaction, Dang Tran, Yingnan Zhu, Michael R. LOVELACE, Xing Zheng, Daniel P. Gicklhorn, Jaffar Khoshgozaran. Samsung Electronics Co. LTD. (Filed)

- Time and Location Based Information Search and Discovery, Alireza Mojtahedi, Ali Khoshgozaran, Amir Raminfar. Tilofy Inc. (Filed)

- Collaborative Decision-Making For Deriving Micro-Recommendations For Offsite Users, Dang Van Tran, Xing Zheng, Jaffar Khoshgozaran, Yingnan Zhu. Samsung Electronics Co. LTD. (Filed)

- Micro-App Dynamic Revenue Sharing, Dang Van Tran, Xing Zheng, Jaffar Khoshgozaran, Yingnan Zhu. Samsung Electronics Co. LTD. (Filed)

- Collaborative Decision-Making For Deriving Micro-Recommendations For Offsite Users, Dang Van Tran, Xing Zheng, Jaffar Khoshgozaran, Yingnan Zhu, Samsung Electronics Co. LTD. (Filed)

MAGAZINE PUBLICATIONS

- Cyrus Shahabi, Farnoush Banaei-Kashani, Ali Khoshgozaran, Luciano Noceraand Songhua Xing, GeoDec: GeoDec: A framework to visualize and query geospatial data for decision-making. IEEE MultiMedia 17(3): 14-23 (2010).

BOOK CHAPTERS

- Ali Khoshgozaran and Cyrus Shahabi, Private Information Retrieval Techniques for Enabling Location Privacy in Location-Based Services, Privacy in Location-based Applications, ISBN: 978-3-642-03510-4, Bettini, C.; Jajodia, S.; Samarati, P.; Wang, X.S. Eds., October 2009, 59-83.

JOURNAL PAPERS

- Ali Khoshgozaran, Houtan Shirani-Mehr and Cyrus Shahabi, Blind evaluation of location based queries using space transformation to preserve location privacy, Geoinformatica Journal, Nov 2012.

- Ali Khoshgozaran, Ali Khodaei, Mehdi Sharifzadeh, Cyrus Shahabi, A Hybrid Aggregation and Compression Technique for Road Network Databases, Knowledge and Information Systems Journal (KAIS) 17(3), 2008.

- Ali Khoshgozaran, Cyrus Shahabi and Houtan Shirani-Mehr, Location Privacy; Going Beyond K-anonymity, Cloaking and Anonymizers, Knowledge and Information Systems Journal (KAIS), 2010.

- Ali Khoshgozaran and Cyrus Shahabi, A taxonomy of approaches to preserve location privacy in location based services, to appear in International Journal of Computational Science and Engineering.

CONFERENCE PAPERS

- Ali Khoshgozaran and Cyrus Shahabi, Towards Private Navigation of Tree Structured Spatial Indexes, The Third International Conference on Emerging Databases (EDB 2011), Incheon, Korea.

- Ali Khoshgozaran and Cyrus Shahabi, Private Buddy Search: Enabling Private Spatial Queries in Social Networks, Symposium on Social Intelligence and Networking (SIN09). In conjunction with IEEE International Conference on Social Computing (SocialCom09), Vancouver, Canada, August 2009, 166-173.

- Luciano Nocera, Arjun Rihan, Songhua Xing, Ali Khodaei, Ali Khoshgozaran, Farnoush Banaei Kashani, Cyrus Shahabi: GeoDec: a multi-layered query processing framework for spatio-temporal data. ACM GIS 2009: 546-547.

- Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi and Kian-Lee Tan, Private Queries in Location Based Services: Anonymizers are not Necessary, In Proceedings of ACM SIGMOD Conference 2008, Vancouver, Canada, 121-132.

- Ali Khoshgozaran, Cyrus Shahabi, Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy, 10th International Symposium on Spatial and Temporal Databases (SSTD), Boston, United States July 2007, 239-257.

- Cyrus Shahabi, Yao-Yi Chiang, Kelvin Chung, Kai-Chen Huang, Jeff Khoshgozaran-Haghighi, Craig Knoblock, Sung Chun Lee, Ulrich Neumann, Ram Nevatia, Arjun Rihan, Snehal Thakkar, Suya You, GeoDec: Enabling Geospatial Decision Making, IEEE International Conference on Multimedia & Expo (ICME), Toronto,Canada, July 2006, 93-96.

WORKSHOP PAPERS

- Ali Khoshgozaran, Enabling Location Privacy; Moving beyond K-anonymity, Cloaking and Anonymizers, ACM SIGMETRICS 2008 Student Thesis Panel, in conjunction with ACM SIGMETRICS’08, June 3-5 2008, Annapolis MD.

- Ali Khoshgozaran, Houtan Shirani-Mehr and Cyrus Shahabi, SPIRAL: A Scalable Private Information Retrieval Approach to Location Privacy, The 2nd International Workshop on Privacy-Aware Location-based Mobile Services (PALMS), In conjunction with the 9th International Conference on Mobile Data Management (MDM'08), April 27 2008, Beijing, China.

- Ali Khoshgozaran, Ali Khodaei, Mehdi Sharifzadeh, Cyrus Shahabi, A Multi-Resolution Compression Scheme for Efficient Window Queries over Road Network Databases, SSTDM 2006, Hong Kong, China, 355-360.

- Cyrus Shahabi and Ali Khoshgozaran, Location Privacy in Geospatial Decision-Making, Invited paper for DNIS 2007, Aizu, Japan, 1-15.

RESEARCH COMMUNITY SERVICE

- Reviewer for the IEEE Transactions on Knowledge and Data Engineering (TKDE)

- Reviewer for International Conference on Very Large Databases (VLDB)

- Reviewer for International Conference on Spatial and Temporal Databases (SSTD)

- Reviewer for IEEE International Conference on Data Engineering (ICDE)

- Reviewer for Special Interest Group on Knowledge Discovery and Data Mining (SIG KDD)

- Reviewer for GeoSensor Networks (GSN) Conference

- Reviewer for International Journal on Advances of Computer Science for Geographic Information Systems (Geoinformatica)

- Reviewer for ACM International Conference on Design of Communication (SIG DOC)

- Reviewer for International Symposium on Web and Wireless Geographical Information Systems (W2GIS)

- Reviewer for International Conference on Mobile Data Management (MDM)

- Reviewer for International Conference on Emerging Databases (EDB)

- Reviewer for ACM Annual Conference on Multimedia Systems

- Reviewer Ubiquitous Data Mining Conference (UDM)

INVITED TALKS/PANELS

- Judge, Demo Day, Object Oriented Design and Programming Course, University of Cal Poly Pomona (2013)

- Keynote Speaker, Iranian American Society of Engineers and Architects, Irvine, CA (2011)
- Panelist, Innovation Forum, Annenberg School of Communication & Journalism, USC (2014)
- Guest Lecturer, Specification and Design of User Interface Software Course, USC (2014)
- Guest Lecturer, Database Systems, USC (2011)
- Guest Lecturer, Data Informatics Professional Practicum, USC (2015)
- Speaker, The Montgomery Summit, Santa Monica, CA (2015)
- Keynote Speaker, Korean Investors & Traders Association Forum, Los Angeles, CA (2016)
- Panelist, Summit in Media & Tech, Hollywood Production Center, Los Angeles, CA (2017)
- Guest Speaker, Shanghai Investor Connect Road Show (2017)
- Guest Speaker, Plug and Play CHTA Disruptive Exhibition (2017)
- Guest Speaker, Integrated Media Services Center Retreat (2017)

PRESS

- How Artificial Intelligence Is Changing Storytelling, Huffington Post
- Innovation in Action: Impact Through Storytelling and Partnerships, United Nations System Staff College
- Perspective 17: A look into the future, Prensario International
- Tilofy Helps Marketers Spot Trends Before They Arise, American Marketing Association
- "First automated trend forecasting platform" predicted the Rainbow Bagel, Marketing Land
- Six Batches – One Mission: Change the Face of Retail Forever, PR Newswire
- USC engineering students find success in startup lab, ABC7
- Start-up companies get shot at big break with Launch Festival 2015, Fox KTVU
- A Day in the Life of the USC Viterbi Startup Garage, USC Viterbi
- Tech Startups & the LA Economy, Vimeo
- Announcing the 2014 socalTECH 50: Southern California's Ones to Watch, SocalTech
- Seven Startups You Probably Haven't Heard Of – But Should, Forbes

- Tilofy brings the power of location based browsing, Digital Journal
- Making Sense Of The World Around You Using Location Data, SocalTech
- Urban Exploration? There's an App for That, USC Viterbi
- Virtual World Meets the Real World, NBC News

AWARDS AND HONORS

- University Fellowship, The George Washington University, CS Dep. 2003-04
- Outstanding Leadership Award, University of Southern California, 2005
- Graduate Research Scholarship, University of Southern California, 2005
- Outstanding Graduate Student Award, APSIH, 2006
- Youth Appreciation Award, NIPOC, 2007
- Yahoo! Ambassador for USC, 2007-2008
- Over 1000 citations to peer reviewed publications, 2005-2010
- Winner: #1 Big Data Startup, USC Startup Garage Competition (100+ participants), 2013
- SocalTech50: Southern California's Top 50 People to Watch in Tech Industry (along with CEOs of Tinder, Snapchat & Whsiper), 2014
- Winner: #1 Disruptive Startup in Retail (800+ participants), Plug & Play Tech Center, 2016
- Winner: #1 Disruptive Media Startup, Plug & Play Tech Center, 2017

EXHIBIT B: LIST OF MATERIALS CONSIDERED

- Complaint filed by the Plaintiff (Case 8:17-cv-00772 Document 1 Filed 05/02/17).
- Answer to Complaint and Counterclaim against Plaintiff filed by Defendant (Case 8:17-cv-00772-CJC-JDE Document 8 Filed 07/05/17).
- Artissimo Designs LLC's Responses to Neofonie GMBH's First Set of Interrogatories filed by Defendant.
- Document titled "artdesigns.com eCommerce Solution" dated Feb 22, 2016.
- Document titled "Offer: Development, Test and Launch of a MVP for artdesigns.com" signed by Contractor (Plaintiff) and Client (Defendant) on May 23 and May 19 of 2016, Respectively.
- JIRA tickets created for the project and their file attachments made available to me at https://jira.neofonie.de/browse/
- Confluence wiki pages and their file attachments made available to me at https://confluence.neofonie.de/display/AR
- Portion of email exchanges between employees of Defendant and Plaintiff.
- Portion of invoices and their task breakdown charts sent to Defendant by Plaintiff.
- Books, publications and papers cited throughout this report.

PROOF OF SERVICE

STATE OF CALIFORNIA, COUNTY OF LOS ANGELES

I am employed in the County of Los Angeles, State of California. I am over the age of 18 and not a party to the within action. My business address is 1875 Century Park East, Suite 2150, Los Angeles, California 90067.

On September 18, 2018, I served the foregoing documents described as: **DISCLOSURE OF EXPERT TESTIMONY PURSUANT TO RULE 26(a)(2)(A), F.R.Civ.P.** on interested parties as follows:

SEE ATTACHED SERVICE LIST

___ **By United States Mail.** I enclosed the documents in a sealed envelope or package addressed to the person above.

___ By depositing the sealed envelope with the United States Postal Service, with the postage fully prepaid.

___ I placed the envelope for collection and mailing, following our ordinary business practices. I am readily familiar with this business's practice for collecting and processing correspondence for mailing. On the same day that the correspondence is placed for collection and mailing, it is deposited in the ordinary course of business with the United States Postal Service, in a sealed envelope with postage fully prepaid.

I am a resident or employed in the county where the mailing occurred. The envelope or package was placed in the mail at Los Angeles, California.

___ **By overnight delivery.** I enclosed the documents in an envelope or package provided by an overnight delivery carrier and addressed to the persons listed above. I placed the envelope or package for collection and overnight delivery at an office or a regularly utilized drop box of the overnight delivery carrier.

___ **By messenger service.** I served the documents by placing them in an envelope or package addressed to the person(s) listed above and provided them to a professional messenger service for service. (*A declaration by the messenger must accompany this Proof of Service or be contained in the Declaration of Messenger below*).

___ **By fax transmission.** Based on an agreement of the parties to accept service by fax transmission, I faxed the documents to the person at the fax numbers listed above. No error was reported by the fax machine that I used. A copy of the record of the fax transmission, which I printed out, is attached.

X **By e-mail or electronic transmission.** Based on a court order or an agreement of the parties to accept service by e-mail or electronic transmission, I caused the documents to be sent to the person at the e-mail addresses listed above. I did not receive, within a reasonable time after the transmission, any electronic messenger or other indication that the transmission was unsuccessful.

X **STATE** I declare under penalty of perjury under the laws of the State of California that the above is true and correct.

____ **FEDERAL** I declare that I am employed in the office of a member of the bar of this court at whose direction this service was made.

Executed on September 18, 2018 at Los Angeles, California.

____
Sacha Emanuel

SERVICE LIST

Roger G. Jones
Peter C. Sales
Kimberly M. Ingram
Bradley Arant Boult Cummings LLP
1600 Division Street, Suite 700
Nashville, TN 37203

Attorneys for Defendant/Counterclaimant
Artissimo Designs LLC